شبكات الحاسب والإنترنت

(الجزء الأول: أسس ومبادئ الشبكات والإنترنت)

تأليف

د. جيمس كيروز

قسم علوم الحاسب

جامعة ماسوشوستس

الولايات المتحدة الأمريكية

د. كيث روس

قسم علوم الحاسب

معهد العلوم التطبيقية بجامعة نيويورك

الولايات المتحدة الأمريكية

نقله إلى العربية

د. السيد محمد الألفي

كلية هندسة وعلوم الحاسب

جامعة الملك فهد للبترول والمعادن

المملكة العربية السعودية

د. رضوان السعيد عبدالعال

كلية هندسة وعلوم الحاسب

جامعة الملك فهد للبترول والمعادن

المملكة العربية السعودية

الطبعة الأولى 2010

ⓒ مكتبة العبيكان، 1431هـ

فهرسة مكتبة الملك فهد الوطنية أثناء النشر

كيروز، جيمس

شبكات الحاسب والإنترنت. / جيمس كيروز؛ كيث روس؛

السيد محمد الألفي. رضوان السعيد عبدالعال. — الرياض، 1431هـ

2 مج.

856 ص، 16.5×24 سم

ردمك: 5-938-54-9960-978 (مجموعة)

: 2-939-54-9960-978 (ج1)

1- شبكات الحاسب 2- الإنترنت أ. روس، كيث (مؤلف مشارك)

ب. الألفي، السيد محمد (مترجم). عبدالعال، رضوان السعيد (مترجم مشارك) ج. العنوان

ديوي: 004.65 1431/48هـ

رقم الإيداع: 1431/48

ردمك: 5-938-54-9960-978 (مجموعة)

: 2-939-54-9960-978 (ج1)



صدر هذا الكتاب بدعم من جامعة الملك فهد للبترول والمعادن تحت مشروع ترجمة كتاب رقم AR060006 وضمن اتفاقية نشر خاصة بين شركة العبيكان للأبحاث والتطوير وعمادة البحث العلمي في الجامعة

الطبعة الأولى

1431هـ/2010م

الناشر: العبيكان للنشر

الرياض — شارع العليا العام — جنوب برج المملكة

هاتف: 2937574/2937581 فاكس: 2937588

ص.ب: 67622 الرمز 11517

التوزيع: مكتبة العبيكان

الرياض — العليا — تقاطع طريق الملك فهد مع العروبة

هاتف: 4654424/4160018 فاكس: 4650129

ص.ب: 62807 الرمز 11595



تصدير

يسعدنا أن نقدم للقارئ العربي هذا الكتاب وهو ترجمة ليست حرفية مع الحفاظ على الدقة في النقل لواحد من أكثر الكتب انتشاراً في الجامعات العالمية في مجال شبكات الحاسب والإنترنت. ويأتي هذا الجهد انطلاقاً من ثقتنا بأن اللغة العربية قادرة على مواكبة التقدم العلمي والتقني المستمر في شتى مجالات الحياة، وأنها لغة ثرية تمتلك الآليات التي تجعلها في مستوى غيرها من اللغات الأخرى كالإنجليزية ولغات شرق أسيا واللغات الأوربية لتدريس المناهج العلمية والتقنية في الجامعات. كما نأمل في أن يكون هذا الكتاب إضافة تثري المكتبة العربية بواحد من أهم الكتب في هذا المجال على المستوى الأكاديمي. وقد قمنا بتعريب المصطلحات اعتماداً على خبرتنا في هذا المجال وبعد الرجوع لمجامع اللغة العربية في العالم العربي وخاصة مجمع اللغة العربية بالقاهرة (معجم المصطلحات العلمية)، والعديد من القواميس والمعاجم والموسوعات العلمية المتخصصة الأخرى (كقاموس المورد، ومعجم الكيلاني لمصطلحات الكمبيوتر والإنترنت، ودليل شعاع لمصطلحات الحاسب، ومعجم مصطلحات الكمبيوتر - الدار العربية للعلوم)، ومكتب تنسيق التعريب بالرباط (بنك المصطلحات الموحدة)، والبنك الآلي السعودي للمصطلحات العلمية (باسم)، والجمعية الدولية للمترجمين واللغويين العرب (واتا)، وشبكة صوت العربية، والمركز العربي للتعريب والترجمة والنشر بدمشق (مجلة التعريب)، وغيرها من مواقع الإنترنت (انظر الملحق 3). ومن الجدير بالذكر أنه رغم وجود الكثير من الجدل حول تعريب العلوم وغرابة المصطلحات واختلافها من بيئة لأخرى إلا أن هذا لا يقلل أهمية الجوانب الإيجابية للترجمة والتعريب.

والله نسأل أن ييسر الإفادة من هذا الكتاب، وأن يكون حافزاً للعديد من المتخصصين، للقيام بأدوار مماثلة لنقل وتوطين العلوم الحديثة وتطبيقاتها في العالم العربي.

المترجمان،،،

شكر وتقدير

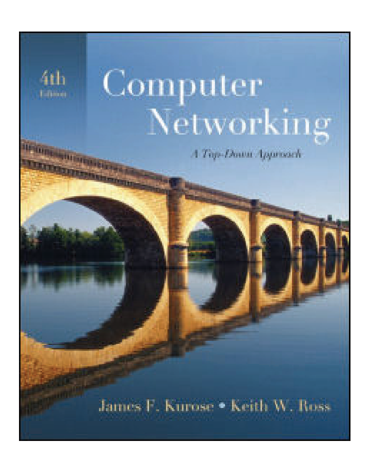
من فضل اللّٰه علينا أن وفقنا لهذا العمل ويسره لنا، فالحمد اللّٰه الذي بنعمته تتم الصالحات، ونستغفره على أي تقصير كان منا.

ونتوجه بخالص الشكر والعرفان لجامعة الملك فهد للبترول والمعادن بالمملكة العربية السعودية، لما أولته من دعم ملحوظ لهذا المشروع، ولدورها الريادي وتشجيعها لكل من يسهم في نهضة المجتمع وخدمته على الصعيد الإقليمي والعربي. كما نتوجه بالشكر لكل من تعاون معنا لإنجاز هذا العمل، ونخص بالذكر معالي مدير الجامعة، وعمادة البحث العلمي، وعميد كلية علوم وهندسة الحاسب، ورئيس قسم علوم الحاسب والمعلومات، ورئيس قسم هندسة الحاسب، والزملاء من أعضاء هيئة التدريس بالكلية.

ونخص بالشكر الدكتور/ عبد اللّٰه بن عمر الحاج إبراهيم الأستاذ المشارك بقسم الدراسات الاسلامية والعربية بجامعة الملك فهد للبترول والمعادن لمراجعة الكتاب لغوياً، والشكر موصول للمحكمين الدوليين (سواء لمقترح المشروع أو لنسخة الكتاب النهائية) على تعليقاتهم البناءة وعلى إشادتهم بالدور الإيجابي للمترجمين في بساطة ووضوح العرض وتسلسل الأفكار.

المترجمان،،،

بيانات الكتاب المُترجَم



- **عنوان الكتاب باللغة الإنجليزية:**

Computer Networking: A Top-Down Approach

- **المؤلفان:** James F. Kurose & Keith W. Ross
- **رقم الإصدار:** الرابع
- **سنة النشر:** 2008
- **الناشر:** Addison Wesley
- **موقع الكتاب على شبكة الويب:**

http://www.aw-bc.com/kurose-ross/

- **مجال التخصص**

العام: هندسة وعلوم الحاسب   الدقيق: شبكات الحاسب

نبذة عن المترجمين

لم يقتصر دور المترجمين على تحويل النص من اللغة الإنجليزية إلى اللغة العربية، ولكنهما اضطلعا بدور آخر – باعتبارهما متخصصين في مجال هندسة وعلوم الحاسب – وهو تنقيح المادة العلمية بما يتناسب مع السياق والبيئة العربية. وفيما يلي نبذة مختصرة عن مؤهلاتهما، وللمزيد من المعلومات يرجى الاطلاع على السيرة الذاتية للباحثين من خلال موقعيهما على شبكة المعلومات الدولية (الويب):

الدكتور/ السيد محمد الألفي:

حاصل على درجة الدكتوراه في هندسة الحاسبات (النظم الذكية وتطبيقاتها في شبكات الحاسب) من معهد ستيفنز للتكنولوجيا بالولايات المتحدة الأمريكية. قام بالتدريس وبإعداد مذكرات المحاضرات لعدد من المناهج في مجال شبكات الحاسب (باللغتين الإنجليزية والعربية). كما ألف عدداً من الأوراق البحثية في هذا المجال، وشارك في اللجان الفنية لعدة مؤتمرات علمية دولية كمؤتمرات IEEE: ICC وGlobecom وWiMob وغيرها. وشارك بفعالية (كرئيس لجنة بقسم علوم الحاسب والمعلومات بجامعة الملك فهد للبترول والمعادن) في تقييم وتحديث البرنامج الدراسي للقسم وإعداد مناهج تفصيلية في مجال شبكات الحاسب في ضوء أحدث توجيهات للجنة الأمريكية (ABET/CSAB) لاعتماد البرامج الأكاديمية في مجال الحاسب الآلي.

البريد الإلكتروني: alfy@kfupm.edu.sa
موقع الويب: http://faculty.kfupm.edu.sa/ics/alfy

الدكتور/ رضوان السعيد عبد العال:

حاصل على درجة الدكتوراه في الهندسة الكهربية من جامعة استراثكليد بالمملكة المتحدة. قام بتدريس هندسة اتصالات البيانات والحاسب بقسم هندسة الحاسب الآلي بجامعة الملك فهد للبترول والمعادن عدة مرات. وقبل ذلك وأثناء عمله بمعهد البحوث بالجامعة كان مسؤولاً عن تحرير الجزء الخاص بإدارة مصادر الطاقة في التقرير السنوي لمعهد البحوث وترجمته إلى اللغة العربية لعدة سنوات. كما قام بترجمة العديد من الوثائق والنشرات الفنية ومستخلصات الأبحاث إلى اللغة العربية، وكذلك ترجمة عدة مقالات علمية منها مقال عن الإنترنت لمجلة العلوم، وهي الترجمة العربية لمجلة ساينتفيك أميريكان الأمريكية وتصدرها مؤسسة الكويت للتقدم العلمي.

البريد الإلكتروني: radwan@kfupm.edu.sa

موقع الويب: http://faculty.kfupm.edu.sa/coe/radwan

جدول المحتويات

☆ جدول محتويات الجزء الأول ☆

الموضوع	الصفحة
• مقدمة	xix
الفصل الأول: مقدمة عن شبكات الحاسب والإنترنت	1
• ما هي الإنترنت؟	3
• حافة الشبكة	16
• قلب الشبكة	36
• التأخير، والفقد، والطاقة الإنتاجية في شبكات تحويل الرزم	53
• طبقات البروتوكولات ونماذج الخدمة الخاصة بها	72
• أمن الشبكات	85
• تاريخ شبكات الحاسب والإنترنت	94
• الخلاصة	104
• أسئلة وتمارين وتدريبات الفصل الأول	108
الفصل الثاني: طبقة التطبيقات	123
• مبادئ تطبيقات الشبكة	125
• الويب وبروتوكول HTTP	147
• نقل الملفات باستخدام بروتوكول FTP	174
• البريد الإلكتروني (E-mail)	178
• خدمة دليل الإنترنت لأسماء النطاقات (DNS)	199
• تطبيقات النظائر (P2P)	221
• برمجة مقابس بروتوكول TCP	244
• برمجة مقابس بروتوكول UDP	257
• الخلاصة	267
• أسئلة وتمارين وتدريبات الفصل الثاني	269
الفصل الثالث: طبقة النقل	283
• مقدمة وخدمات طبقة النقل	285
• التجميع والتوزيع	292

- بروتوكول النقل اللاتوصيلي: UDP . . . 303
- أساسيات النقل الموثوق للبيانات . . . 312
- بروتوكول النقل التوصيلي: TCP . . . 347
- مبادىء التحكم في الازدحام . . . 390
- التحكم في الازدحام في بروتوكول TCP . . . 405
- الخلاصة . . . 423
- أسئلة وتمارين وتدريبات الفصل الثالث . . . 428

الفصل الرابع: طبقة الشبكة 445

- مقدمة . . . 448
- شبكات الدائرة الافتراضية وشبكات وحدات البيانات . . . 457
- ماذا بداخل الموجّه؟ . . . 466
- بروتوكول الإنترنت (IP): التمرير والعنونة في الإنترنت . . . 482
- خوارزميات التوجيه . . . 528
- التوجيه في شبكة الإنترنت . . . 557
- توجيه البث الإذاعي (العام) والتوجيه المتعدد (الجماعي) . . . 579
- الخلاصة . . . 598
- أسئلة وتمارين وتدريبات الفصل الرابع . . . 600

الفصل الخامس: طبقة ربط البيانات والشبكات المحلية 617

- مقدمة عن طبقة ربط البيانات وخدماتها . . . 620
- أساليب اكتشاف أخطاء البيانات وتصحيحها . . . 629
- بروتوكولات الوصول المتعدد . . . 639
- العنونة في طبقة ربط البيانات . . . 660
- شبكة الإيثرنت . . . 670
- محوّلات طبقة ربط البيانات . . . 689
- بروتوكول نقطة إلى نقطة (PPP) . . . 698
- الوصلة الافتراضية: الشبكة كطبقة ربط البيانات . . . 705
- الخلاصة . . . 718
- أسئلة وتمارين وتدريبات الفصل الخامس . . . 721

ثبت المراجع 731

الملاحق[1] 767

- الملحق 1: مسرد الاختصارات 767
- الملحق 2: مسرد المصطلحات والتعبيرات 779
- الملحق 3: وقفة مع ترجمة وتعريب المصطلحات 821

[1] مضافة من قِبَل المترجمين

جدول المحتويات

☆ جدول محتويات الجزء الثاني ☆

الموضوع	الصفحة
الفصل السادس : الشبكات اللاسلكية والنقّالة	1

- مقدمة 4
- خصائص الوصلات والشبكات اللاسلكية 10
- الشبكة المحلية اللاسلكية 802.11 (WiFi) 19
- الاتصال الخلوي بالإنترنت 49
- مبادئ إدارة قابلية الحركة 60
- بروتوكول IP النقّال 73
- قابلية الحركة في الشبكات الخلوية 79
- تأثير وصلة اللاسلكي وقابلية الحركة على بروتوكولات الطبقات العليا 87
- الخلاصة 91
- أسئلة وتمارين وتدريبات الفصل السادس 93

الفصل السابع : شبكات الوسائط المتعددة	101

- تطبيقات الوسائط المتعددة على الشبكات 103
- تشغيل بيانات الصوت والفيديو المخزّنة 118
- الاستغلال الأمثل لخدمة أفضل جهد 133
- بروتوكولات للتطبيقات الفورية التفاعلية 157
- توفير فئات متعددة من الخدمة 182
- توفير ضمانات لجودة الخدمة 210
- الخلاصة 221
- أسئلة وتمارين وتدريبات الفصل السابع 224

الفصل الثامن : أمن الشبكات	237

- ماذا يعني أمن الشبكة؟ 239
- مبادئ تشفير البيانات 243
- سلامة الرسائل 263
- التحقّق من النقطة الطرفية 278

- تأمين البريد الإلكتروني 289
- بروتوكول SSL لتأمين توصيلات TCP 298
- أمن طبقة الشبكة: IPsec 307
- تأمين الشبكات المحلية اللاسلكية 313
- الأمن التشغيلي: برامج الحماية وأنظمة كشف الاختراقات 320
- الخلاصة 335
- أسئلة وتمارين وتدريبات الفصل الثامن 338

الفصل التاسع: إدارة الشبكات 345

- مفهوم إدارة الشبكة 346
- البنية التحتية لإدارة الشبكة 353
- إطار إدارة الشبكة بمعيار الإنترنت 360
- معيار ASN.1 382
- الخاتمة 388
- أسئلة وتمارين وتدريبات الفصل التاسع 390

ثبت المراجع 393

الملاحق[2] 429

- الملحق 1: مسرد الاختصارات 429
- الملحق 2: مسرد المصطلحات والتعبيرات 441
- الملحق 3: وقفة مع ترجمة وتعريب المصطلحات 483

[2] مضافة من قِبَل المترجمين

مقدمـــــة

برزت في الآونة الأخيرة أهمية شبكات الحاسب الآلي – وبخاصة الإنترنت – وأحدثت تغييرات جوهرية على مستوى الأفراد والمجتمعات، فقد زاد اعتماد الأفراد والشركات والجامعات والهيئات الحكومية وغير الحكومية على الخدمات التي تقدمها تلك الشبكات كخدمة الاتصال عن طريق البريد الإلكتروني، والمحادثة الفورية بين الأفراد، وغيرها. كما غيرت طريقة نشر وتبادل المعلومات كالمكتبات الرقمية، ومع التطور المستمر استُحدِثت العديد من التطبيقات لزيادة الإفادة من الشبكة كالتعليم عن بعد، والحكومة الإلكترونية، والتسوق الإلكتروني، والمؤتمرات عبر الشبكة، وخدمات الترفيه والألعاب من خلال الشبكة. مما أدى إلى ظهور تقنيات حديثة لدعم تلك التطبيقات وتوفير خدمة مقبولة للمستخدم.

ولقد تم نشر العديد من الكتب والمراجع في هذا المجال، منها الكتاب الذي بين أيدينا:

"*Computer Networking: A Top-Down*, 4th Edition. By James F. Kurose and Keith W. Ross, Addison Wesley 2008".

ويُعد هذا الكتاب من أكثر الكتب الأكاديمية انتشاراً في مجال شبكات الحاسب، حيث يُستخدم في المئات من الكليات والجامعات العالمية المرموقة كمرجع أساسي أو موصى به؛ من بينها:

- Harvard University
- University of California at Berkeley
- Massachusetts Institute of Technology (MIT)
- California Institute of Technology
- Princeton University
- University of Texas at Austin
- Columbia University
- McGill University
- Cornell University
- Massachusetts University
- Purdue University

- Georgia Institute of Technology
- Boston University
- Texas A&M University
- University of Massachusetts at Amherst
- University of California at Riverside
- Rensselaer Polytechnic Institute
- University of Houston
- University of Michigan
- University of Texas at San Antonio
- University of New South Wales
- University of Florida
- King Fahd University of Petroleum and Minerals

ولقد تُرجم هذا الكتاب إلى أكثر من عشر لغات، ويستخدمه أكثر من مئة ألف طالب وممارس في العالم، وقد أثنى الكثيرون ممن قرؤوا الكتاب عليه وأرسلوا ردود فعل إيجابيه للمؤلفين؛ من بينهم:

- Leonard Kleinrock, University of California at Los Angeles (The father of the Internet)
- Tim-Berners Lee, World Wide Web Consortium (Web Inventor)
- Shivkumar Kalyanaraman, Rensseler Polytechnic Institute
- Jennifer Rexford, AT&T Labs
- Vinton Cerf, Google
- Radia Perlman, Sun Microsystems
- Sally Floyd, University of California at Berkeley
- Mario Gerla, University of California at Los Angeles
- Jennifer Rexford, Princeton University
- Al Aho, Columbia University
- Pratima Akkunoor, Arizona State University
- Willis Marti, Texas A&M University
- Paul Amer, University of Delaware
- Sally Floyd, University of California at Berkeley
- David Kotz, Dartmouth College
- Yechiam Yemini, Columbia University
- Don Towsley, University of Massachusetts
- Phil Zimmermann, independent consultant
- Mischa Schwartz, Columbia University
- Bob Metcalfe, International Data Group

يعتقد مؤلفا الكتاب أن ذلك النجاح الباهر للكتاب يرجع إلى أنه قَدَّم – منذ طبعته الأولى في عام 2000 – أسلوباً جديداً وفريداً لمعالجة شبكات الحاسب يختلف عن الطرق المتبعة في معظم المراجع الأخرى.

فبعد مقدمة شاملة يعطي الكتاب فكرة مبسطة عن شبكات الحاسب، ثم يتناول قضايا طبقة التطبيقات أو البرامج بمزيد من التفصيل – مع أمثلة حية من شبكة الإنترنت – قبل الاستفاضة في تفاصيل الطبقات الأدنى. ولهذا الأسلوب عدة فوائد مهمة في مجال التدريس، منها التدرج في عرض الموضوع بدءاً من الأسهل تصوراً إلى الأصعب، ومن الإجمال إلى التفصيل، ومن خدمات الشبكة (وهي برامج الشبكة التي يكون القارئ قد تعرض لها غالباً من قبل، كالتصفح على شبكة الويب، واستخدام البريد الإلكتروني، والمحادثة عبر الشبكة) وانتقالاً إلى الطبقات التي تدعم تلك الخدمات.

كما أن المعالجة المبكرة لنماذج طبقة التطبيقات وبيئات البرمجة يعطي القارئ فرصة أكبر للممارسة العملية لمفاهيم وبروتوكولات الشبكة المختلفة – التي سيتعامل معها بعد ذلك – في سياق برامج الشبكات، وهو أسلوب تحفيزي قوي للطلاب، فالطلاب يتشوقون أكثر لمعرفة كيفية عمل برامج الشبكات وطريقة تطويرها، قبل إغراقهم بتفاصيل قد يصعب تخيلها للوهلة الأولى. كما أن العديد من التطورات قد استحدثت في تلك الطبقة كشبكة الويب، ومشاركة الملفات بين النظائر، والتشغيل المستمر للوسائط المتعددة، والتسوق من خلال الشبكة، والتعليم عن بعد، والحكومة الإلكترونية، وغيرها.

وتعتبر الطبعة الرابعة آخر ما صدر من الكتاب، وهي طبعة منقحة تعكس التطورات الحديثة في مجال شبكات الحاسب. ومن السمات الإضافية لتلك الطبعة تحديث للمراجع المشار إليها في كل فصل، وتركيز أكثر على موضوعات أمن الشبكة، وتحديث لتقنيات الشبكات اللاسلكية والنقالة، وتنقيح للجزء الخاص بشبكات النظائر وتوزيع المحتوى، كما تحتوي تلك الطبعة على مجموعة جديدة من التجارب العملية تتضمن تدريبات على البرمجة واستخدام برنامج Ethereal.

وللكتاب موقع على شبكة الويب يتضمن العديد من المواد العلمية، والأشكال التوضيحية، والاختبارات القصيرة التفاعلية، وعروض بوربوينت (PowerPoint) لكل فصل من فصول الكتاب، وقائمة بالمراجع، ودليل الحلول للمعلم.

وينقسم الكتاب إلى تسعة فصول – بالإضافة إلى قائمة المراجع – فيما يلي بيانها:

- **الفصل الأول**: يقدم استعراضاً ملخصاً شاملاً للتعريف بشبكات الحاسب والإنترنت، حيث يقدم المفاهيم والمصطلحات الفنية الأساسية تمهيداً لبقية فصول الكتاب. كما يتناول تعريفاً بمكونات وخدمات الشبكة ومبادئ البنية المعمارية وطبقات البروتوكولات، ويحتوي على تأريخ مبسط لشبكات الحاسب والإنترنت.
- **الفصل الثاني**: يناقش خصائص تطبيقات الشبكات وكيفية تطويرها. فبعد تعريف المفاهيم الرئيسة لتطبيقات الشبكات – كبروتوكولات طبقة التطبيقات، ومفهوم الخادم والزبون، ومفهوم برمجة المقابس – يتناول بمزيد من التفصيل العديد من تطبيقات الشبكة كالويب، والبريد الإلكتروني، ونظام خدمة أسماء النطاقات، ومشاركة الملفات عبر شبكة النظائر. بعد ذلك يستعرض كيفية تطوير برامج الشبكة مع مجموعة من الأمثلة المبسطة بلغة البرمجة Java.
- **الفصل الثالث**: يركز على مبادئ وبروتوكولات طبقة النقل (Transport Layer) وكيفية عملها لتوصيل رسائل طبقة التطبيقات. كما يناقش طرق التوصيل المختلفة والإجراءات المطلوبة لضمان توصيل الرسائل بطريقة سليمة عبر الشبكة، كإجراءات التحكم في الأخطاء، والتحكم في تدفق البيانات، وتفادي الازدحام عبر الشبكة.
- **الفصل الرابع**: يتناول طبقة الشبكة (Network Layer) والتي تمثل العمود الفقري لشبكة الحاسب وهي من أعقد الطبقات، حيث يناقش طرق

وبروتوكولات التوجيه المختلفة، وطرق العنونة لتعريف أجهزة الشبكة بعضها ببعض.

- **الفصل الخامس**: يستعرض مبادئ وبروتوكولات طبقة ربط البيانات (Data Link Layer) المختلفة مثل PPP وATM وMPLS، كما يناقش أساليب مشاركة الوسط الناقل وتقنيات الشبكات المحلية ((Local Area Networks (LANs).

- **الفصل السادس**: يناقش الخصائص المميزة والتحديات الناجمة عن الطبيعة الخاصة للوصلات اللاسلكية، كما يتناول البنية المعمارية وبروتوكولات الاتصال المختلفة للشبكات اللاسلكية والنقالة ومبادئ وطرق إدارة التجوال. كذلك يستعرض التأثيرات المترتبة على استخدام الوصلات اللاسلكية والتجوال على بروتوكولات طبقة النقل وطبقة التطبيقات.

- **الفصل السابع**: يتناول تطبيقات الوسائط المتعددة ومتطلباتها وكيفية دعم تلك التطبيقات على شبكة الإنترنت لتوفير خدمة ذات جودة مقبولة.

- **الفصل الثامن**: يقدم تعريفاً بأمن الشبكات وطرق الهجوم على موارد الشبكة والطرق المختلفة لتأمينها وحمايتها من الاستخدام غير المرخص، كما يتناول بروتوكولات الأمن في الطبقات المختلفة وطرق توزيع مفاتيح التشفير وتأمين الشبكات اللاسلكية.

- **الفصل التاسع**: يتناول مفاهيم إدارة الشبكة والبنية المعمارية وبروتوكولات وقواعد المعلومات المستخدمة لمساعدة المشرف على الشبكة في مراقبة مكونات الشبكة وتحليل أدائها لاتخاذ الإجراءات الوقائية الكفيلة بتفادي الأعطال المحتملة وتغيير الإعدادات لتحسين الأداء وتحقيق الاستغلال الأمثل لموارد الشبكة، وكذلك تحليل الأعطال وأسبابها وطرق علاجها، والتحكم في حسابات وصلاحيات مستخدمي الشبكة.

يعمل **المؤلف الأول للكتاب** – وهو الدكتور جيمس كيروز – حالياً أستاذاً بقسم علوم الحاسب في جامعة ماسوشوستس بالولايات المتحدة الأمريكية (وكان رئيساً سابقاً له). وقد حصل على درجة الدكتوراه في علوم الحاسب من جامعة كولومبيا، كما يعمل مديراً مشاركاً لمختبر أبحاث الشبكات، ومديراً مشاركاً لمركز البحوث الهندسية للاستشعار التكيفي التعاوني للأرصاد الجوية بالهيئة القومية الأمريكية للبحوث العلمية (NSF). وتتضمن اهتماماته البحثية البروتوكولات والبنية المعمارية لشبكات الحاسب، وقياسات أداء الشبكة، وشبكات الاستشعار، واتصالات الوسائط المتعددة، ونمذجة وتقييم الأداء.

ولقد عمل الدكتور كيروز رئيساً لتحرير مجلة وقائع الاتصالات (Communications Transactions) الصادرة عن هيئة IEEE، ورئيساً مؤسّساً لتحرير مجلة وقائع الشبكات (Networking Transactions) الصادرة عن الهيئتين IEEE وACM، وكان أحد المؤسسين لمبادرة الكومنولث لتكنولوجيا المعلومات بولاية ماسوشوستس. كما شارك في العديد من لجان المؤتمرات مثل IEEE Infocom، ACM Sigcomm، ACM Sigmetrics لسنوات عدة، وعمل رئيساً مشاركاً للبرنامج الفني لتلك المؤتمرات. ومُنح العديد من الجوائز مثل: جائزة المعلم المتميز لثماني مرات من الجامعة التقنية الوطنية (NTU)، وجائزة المعلم المتميز من كلية علم الطبيعة والرياضيات في جامعة ماسوشوستس، وجائزة التدريس لعام 1996 للجمعية الشمالية الشرقية للدراسات العليا، وجائزة الارتقاء بأعضاء هيئة التدريس من IBM، ووسام التدريس IEEE Taylor Booth، وزمالة GE، وزمالة Lilly للتدريس.

أما **المؤلف الثاني للكتاب** – وهو الدكتور كيث روس – فيعمل أستاذاً بقسم علوم الحاسب بمعهد العلوم التطبيقية (Polytechnic) بجامعة نيويورك بالولايات المتحدة الأمريكية، وقد حصل على درجة الدكتوراه في علوم الحاسب من جامعة بنسلفانيا. كما أشرف على خمس عشرة رسالة دكتوراه، وقام بنشر العديد من الأبحاث وألّف كتابين، كما شارك في تحرير عدد من المجلات العلمية مثل مجلة وقائع الشبكات (ACM/IEEE Networking Transactions)، وشارك في العديد من المؤتمرات مثل: ACM Sigcomm وIEEE Infocom. وتشمل اهتماماته البحثية شبكات الوسائط المتعددة، وبروتوكولات الشبكات، ونظم النظائر (P2P).

❖ أهمية الكتاب

يعتبر هذا الكتاب من أفضل ما نشر في مجال شبكات الحاسب وأحدثها، ويُستخدم في العديد من جامعات العالم لتعليم طلاب البكالوريوس والدراسات العليا، كما يُستخدم كمرجع أساسي من قبل العديد من الباحثين والعاملين في حقل الشبكات.

❖ مدى الإفادة من الكتاب

إن نشر مرجع أكاديمي مثل هذا الكتاب يمثل إضافة حقيقية للمكتبة العربية، وهو ما يؤكد أهمية الدور الريادي للجامعات في خدمة المجتمع. حيث يساعد ذلك على دعم العملية التعليمية في الجامعات العربية (التي تستخدم اللغة العربية في عملية التدريس)، وفي كليات المجتمع، وكليات المعلمين، وكليات البنات، والكليات التقنية، وكليات علوم الحاسب والمعلومات، وكليات الاتصالات والمعلومات ... إلخ. كما يساعد على دعم البحث العلمي، وتنشيط حركة الترجمة، ومواكبة التقدم العلمي في واحد من المجالات الحيوية ألا وهو مجال شبكات الحاسب.

❖ نوعية القراء

هذا الكتاب من الكتب الأكاديمية الشاملة في مجال شبكات الحاسب والتي يمكن أن يستفيد منها فئات متعددة وبخاصة في المجال الأكاديمي والبحثي كطلاب البكالوريوس والدراسات العليا والباحثين والمحترفين في مجال الشبكات. كما يمكن أن يستفيد منه فئات أخرى من القراء تضم: المهتمين بنقل وتوطين التقنيات الحديثة للعالم العربي، والمثقفين الراغبين في التعلم والتثقيف الذاتي في مجال تقنية المعلومات.

مقدمة عن
شبكات الحاسب والإنترنت

Introduction to Computer Networks and the Internet

محتويات الفصل:

- ما هي الإنترنت؟
- حافة الشبكة
- قلب الشبكة
- التأخير، والفقد، والطاقة الإنتاجية في شبكات تحويل الرزم
- طبقات البروتوكولات ونماذج الخدمة الخاصة بها
- أمن الشبكات
- تاريخ شبكات الحاسب والإنترنت
- الخلاصة

ابتداءً من متصفحات الويب في الهواتف الخلوية إلى المقاهي الموصّلة لاسلكياً بالإنترنت، ومن الشبكات المنزلية بوصول سريع وحيز ترددي عريض إلى البنية المعلوماتية التحتية بمواقع العمل التقليدية حيث يوجد حاسب موصل بالشبكة على كل مكتب، إلى السيارات المشبّكة، إلى المجسّات (أجهزة الاستشعار) البيئية المشبّكة، إلى إنترنت ما بين الكواكب – لاشك في أن شبكات الحاسب موجودة الآن بشكلٍ أساسي في كل مكان، كما يجري تطوير تطبيقات جديدة ومثيرة لتوسيع مدى وصول شبكات اليوم إلى حدود أبعد. سيزوّدك هذا الكتاب بمقدمة حديثة عن المجال الديناميكي لشبكات الحاسب،حيث سيزودك بالمبادئ الأساسية والعملية التي تحتاجها لفهم الشبكات، ليس فقط شبكات اليوم بل وشبكات الغد كذلك.

في هذا الفصل سنقدّم نظرة عامة عن شبكات الحاسب والإنترنت. سوف نغطّي الكثير من الموضوعات في هذا الفصل التمهيدي ونناقش العديد من مكونات شبكات الحاسب بدون إغفال للصورة العامة. لذا فهذا الفصل يضع الأساس لبقيّة فصول الكتاب.

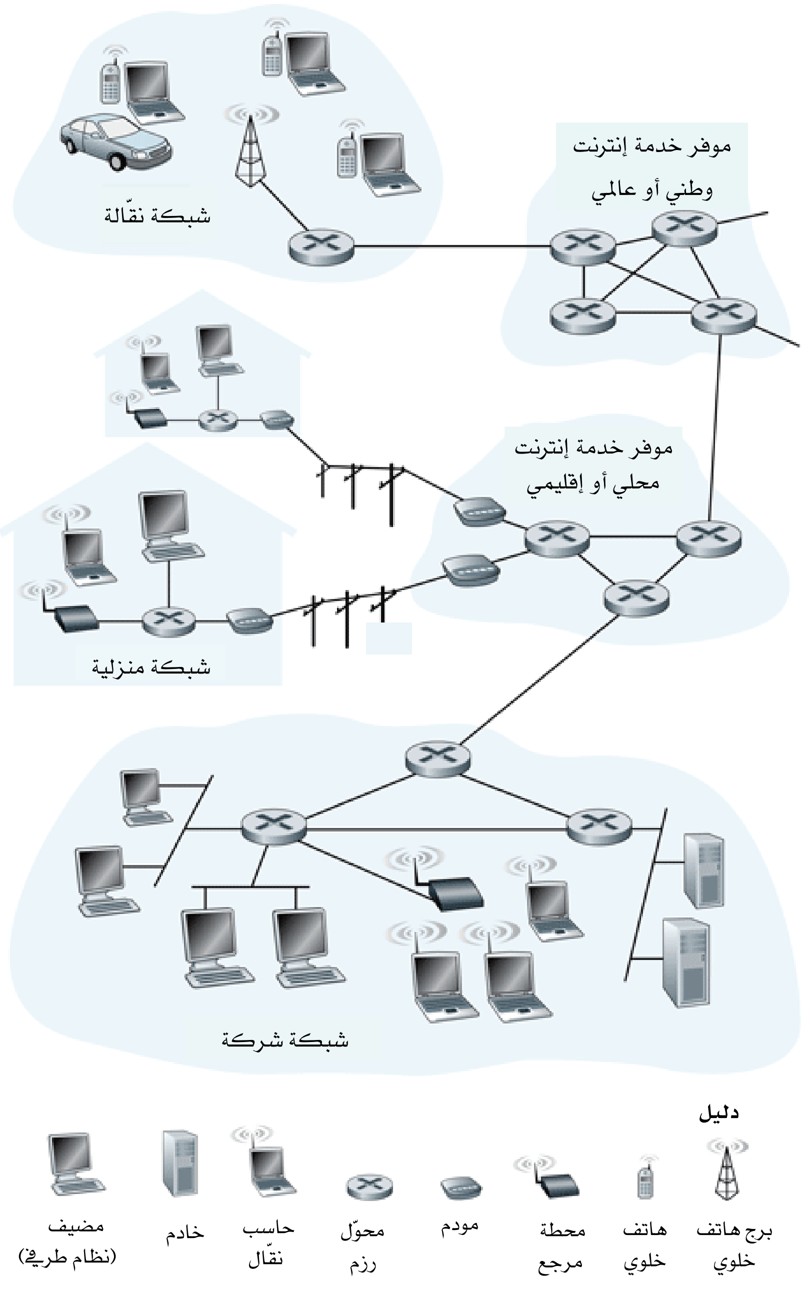
سوف نُنشئ التصور العام لشبكات الحاسب في هذا الفصل كالتالي: بعد تقديم بعض المصطلحات والمفاهيم الأساسية، سنتناول المكونات الأساسية المادية (hardware) والبرمجية (software) التي تشكّل الشبكة، سنبدأ من حواف الشبكة حيث الأنظمة الطرفية (end systems) والتطبيقات الجاري تشغيلها على الشبكة، بعد ذلك سندلف لاستكشاف قلب شبكة الحاسب، وذلك بفحص الوصلات (links) والمحوّلات (switches) التي تنقل البيانات، بالإضافة إلى شبكات الوصول والوسائط المادية التي تربط الأنظمة الطرفية بقلب الشبكة. وسنرى حينئذٍ أن الإنترنت هي شبكة تتألف من شبكات، وسندرك كيف ترتبط تلك الشبكات ببعضها.

بعد استعراض هذه الصورة العامّة لحافة وقلب الشبكة، سنلقي نظرة أكثر تجريداً وشمولاً في النصف الثاني من هذا الفصل، حيث سنتناول اعتبارات التأخير (delay)، والفقد (loss)، والطاقة الإنتاجية (أى معدل تدفق البيانات)

(throughput) في شبكة الحاسب. كما سنقدم نماذج كمّية بسيطة لمعدل التدفق والتأخير من طرف إلى طرف آخذين في الاعتبار التأخير بسبب الإرسال (transmission)، وانتقال الإشارة (propagation)، والانتظار في الصف (الطابور) (queueing). ثم نتناول بعد ذلك بعض المبادئ الأساسية في معمارية شبكات الحاسب، وبالتحديد طبقات البروتوكولات ونماذج تبادل الخدمات بين تلك الطبقات، وسنرى أيضاً كيف أن شبكات الحاسب عرضة لأنواع كثيرة من الهجمات التى سنتناول بعضها بالشرح، ونعرف كيف يمكن جعل تلك الشبكات أكثر أمناً، وأخيراً سنختم هذا الفصل باستعراض مختصر لتاريخ شبكات الحاسب.

1-1 ما هي الإنترنت؟

في هذا الكتاب سنتخذ من الإنترنت العامّة – كشبكة حاسب محددة – وسيلة رئيسية لشرح شبكات الحاسب وبروتوكولاتها. ولكن ما هي الإنترنت؟ كنا نتمنى أن نعطي هنا تعريفاً من جملة واحدة للإنترنت، تعريفاً شاملاً يمكنك أن تأخذه إلى المنزل وتشارك فيه عائلتك وأصدقاءك. ولكن للأسف الإنترنت كائن معقّد للغاية ودائم التغير باستمرار، سواءً من حيث مكوناتها المادية (hardware) والبرمجية (software)، أو من حيث الخدمات التي توفرها. ولذا، فبدلاً من إعطاء تعريف من جملة واحدة، سنحاول إتباع منهج أكثر تفصيلاً. هناك طريقتان للقيام بذلك: تعتمد الطريقة الأولى على وصف تفاصيل ودقائق الإنترنت، أي مكوناتها الأساسية من مكونات مادية وبرمجية. في حين تعتمد الطريقة الثانية على وصف الإنترنت كبنية تحتية من الشبكات التى توفر خدمات للتطبيقات الموزّعة. لنبدأ بوصف تفاصيل ودقائق مكونات الإنترنت، مع الاستعانة بالشكل 1-1 لتوضيح الصورة.



الشكل 1-1 بعض مكونات الإنترنت.

1-1-1 وصف المكونات

الإنترنت شبكة حاسب تربط بين الملايين من الأجهزة الحاسبة في مختلف أنحاء العالم. منذ عهد ليس بالبعيد، كانت تلك الأجهزة الحاسبة تتكون أساساً من حاسبات مكتب شخصية تقليدية أو محطات عمل لاينكس (Linux)، وما يسمّى بالخادمات (servers) التي تخزن وترسل المعلومات كصفحات الويب ورسائل البريد الإلكتروني. ولكن في الآونة الأخيرة تزايد بشكلٍ مضطرد عدد الأنظمة الطرفية غير التقليدية للإنترنت، كالمساعدات الرقمية الشخصية (PDAs)، وأجهزة التليفزيون، والحاسبات النقّالة، والهواتف الخلوية، وكاميرات الويب، والسيارات، والمجسات البيئية، وإطارات الصور، والأنظمة الكهربية والأمنية للمنازل. في الواقع لقد أصبح تعبير "شبكة حاسب" تعبيراً لا يمثل الحقيقة إلى حدٍ ما، لوجود العديد من الأجهزة غير التقليدية الموصلة بالإنترنت الآن. حسب مفردات الإنترنت، يطلق على كل تلك الأجهزة أنظمة طرفية أو مضيفات (hosts). في يوليو/تموز عام 2006 وصل عدد الأنظمة الطرفية التي تستخدم الإنترنت إلى 400 مليون نظام طرفي، وهذا العدد في تزايد مضطرد وسريع [ISC 2007].

يتم توصيل الأنظمة الطرفية سوية بواسطة شبكة اتصال ومحوّلات رزم (packet switches). وسنرى في الجزء 1-2 أن هناك العديد من وصلات الاتصال المصنوعة من أنواع مختلفة من وسائط النقل المادية تضم الكبلات المحورية، والأسلاك النحاسية، والألياف الضوئية، وطيف ترددات الراديو. يمكن للوصلات المختلفة إرسال البيانات بمعدلات مختلفة، حيث يقاس معدل إرسال البيانات على الوصلة بعدد البتات المرسلة فى الثانية (بت/ثانية). عندما يتوافر لدى نظام طرفي بيانات للإرسال إلى نظام طرفي آخر، يقوم نظام الإرسال بتجزئة البيانات وإضافة بايتات الترويسة (header bytes) إلي كل جزء. في مفردات شبكات الحاسب يُطلق على حزم البيانات الناتجة رزم. يتم إرسال الرزم بعد ذلك عبر الشبكة إلى النظام الطرفي المستهدف، حيث يعاد تجميعها من جديد للحصول على البيانات الأصلية.

يتسلم محوّل الرزم الرزمة من إحدى وصلات الاتصال القادمة إليه، ويرسلها عبر إحدى وصلات الاتصال الخارجة منه. وتتخذ محوّلات الرزم أشكالاً مختلفة، لكن النوعين الأبرز في الإنترنت اليوم هما الموجّهات (routers) ومحوّلات طبقة ربط البيانات (link-layer switches). يُرسل كلا النوعين من المحوّلات الرزم باتجاه وجهتها النهائية. وسوف نفحص بالتفصيل الموجّهات في الفصل الرابع ومحوّلات طبقة ربط البيانات في الفصل الخامس. تعرف سلسلة وصلات الاتصال ومحوّلات الرزم التي تعبرها رزمة تخرج من النظام الطرفي المُرسِل إلى النظام الطرفي المُستقبِل بـ "مسار" (route) عبر الشبكة. تعتبر شبكات تحويل الرزم التي استحدثت في السبعينيات من القرن الماضي السلف الأول لإنترنت اليوم. من الصعوبة بمكان تقدير حركة البيانات المنقولة على شبكة الإنترنت على وجه الدقة [Odylsko 2003]، ومع ذلك فقد أعلنت شركة AOL (وهي أحد موفري خدمة الإنترنت) عام 2005 أن حركة بيانات الإنترنت كانت تدخل شبكتها بمعدل 250 جيجابت/ثانية [Gill 2005]. ويقدّر [PriMetrica 2007] أنّ 5 تيرابت/ثانية من سعة حركة البيانات الدولية تم استخدامها من قبل الناقلين العموميين (public carriers) في عام 2006، وأن عرض الحيز الترددي (bandwidth) المستخدم في الاتصالات يتضاعف مرة واحدة كل عامين تقريباً.

تشبه شبكات تحويل الرزم (التي تنقل رزم البيانات) في كثير من جوانبها شبكات المواصلات من الطرق والشوارع والتقاطعات (التي تنقل المركبات). خذ على سبيل المثال مصنعاً يحتاج لنقل كمية كبيرة من البضائع إلى مخزن على بعد آلاف الكيلومترات منه. يتم في المصنع تقسيم البضاعة إلى أجزاء وتحميلها على أسطول من الشاحنات، تنطلق كل شاحنة بشكلٍ مستقل عبر شبكة من الشوارع والطرق السريعة والتقاطعات باتجاه المخزن المستهدف. وفيه يتم تفريغ حمولات الشاحنات وتجميعها مع بعضها. يتضح وجود تشابه بين نقل البيانات ونقل البضائع من عدة جوانب، حيث تشبه الرزم الشاحنات، وتناظر وصلات الاتصال الشوارع والطرق السريعة، وتماثل محوّلات الرزم تقاطعات الطرقات، بينما تشبه الأنظمة الطرفية البنايات (المصنع والمخزن). وتماماً كما تأخذ الشاحنة مسارها عبر شبكة المواصلات، تأخذ رزمة البيانات مسارها عبر شبكة الحاسب.

تتواصل الأنظمة الطرفية مع الإنترنت عن طريق موفري خدمة الإنترنت (ISPs)، بما في ذلك موفري خدمة الإنترنت السكنية كشركة AOL، وشركات الكبل أو الهاتف المحلية، وشركات توفير خدمة الإنترنت للشركات وللجامعات، وكذلك موفري خدمة الإنترنت التي توفر تواصلاً لاسلكياً مع الشبكة في المطارات والفنادق والمقاهي وغيرها من الأماكن العامة كشركة T-Mobile. يعتبر كل موفر لخدمة الإنترنت في حد ذاته شبكة من محوّلات الرزم ووصلات الاتصال. ويقدم موفرو خدمة الإنترنت للأنظمة الطرفية تشكيلة من طرق الوصول للشبكة، بما في ذلك الاتصال عبر خط الهاتف عن طريق المودم بسرعة 64 كيلوبت/ثانية، والوصول المنزلي بحيز ترددي عريض باستخدام مودم كبلي أو وصلة DSL، والوصول عالي السرعة عن طريق شبكة محلية (LAN)، والوصول اللاسلكي. كما يقدم موفرو خدمة الإنترنت طرقاً للوصول للشبكة تمكن مزودي المحتوى من توصيل مواقع الويب مباشرةً إلى الإنترنت. لتحقيق الاتصال ما بين مستخدمي الإنترنت وتمكينهم من الوصول للمحتوى العالمي للشبكة، يتم ربط موفري الخدمة هؤلاء (في المستوى الأدنى) بمزوّدي خدمة الإنترنت الوطنيين والدوليين (في مستوى أعلى)، مثل AT&T وSprint. تتضمن شبكة موفري خدمة الإنترنت في المستوى الأعلى موجّهات سريعة مشبَّكة بوصلات ألياف ضوئية سريعة. تدار كل شبكة من شبكات موفري خدمة الإنترنت، سواء من المستوى الأدنى أو الأعلى، بشكلٍ مستقل وتعمل ببروتوكول IP (انظر أدناه)، كما تتوافق مع عدة أعراف متفق عليها من حيث التسمية والعنونة. سوف نستعرض موضوع موفري خدمة الإنترنت وطرق ربطهم بتفصيل أكثر في الجزء 1-3.

تعمل الأنظمة الطرفية ومحوّلات الرزم وغيرها من مكونات الإنترنت ببروتوكولات تنظم عملية إرسال واستلام المعلومات عبر الإنترنت. ويعتبر بروتوكول التحكم في الإرسال (TCP) وبروتوكول الإنترنت (IP) من أهم بروتوكولات الإنترنت، حيث يحدّد بروتوكول الإنترنت IP صيغة الرزم التي ترسل وتستلم بين الأنظمة الطرفية والموجّهات. يطلق على بروتوكولات الإنترنت الرئيسية

بشكلٍ جماعي مجموعة بروتوكولات TCP/IP (سنبدأ باستعراض البروتوكولات في هذا الفصل التمهيدي، ولكن هذه مجرد بداية فقط، فالجزء الأكبر من هذا الكتاب ينصبّ بشكلٍ أساسي على بروتوكولات شبكات الحاسب).

نظراً للأهمية الخاصة للبروتوكولات بالنسبة للإنترنت، فإنه من الضروري أن يتفق كل المعنيين على كل مايتعلق بعمل كل بروتوكول، وهنا يأتي الدور الهام الذي تلعبه المعايير القياسية. يتم تطوير معايير الإنترنت بواسطة فريق عمل هندسة الإنترنت (IETF) [IETF 2007]. ويطلق على وثائق مواصفات تلك اللجنة "طلبات التعليقات" (RFCs). بدأت تلك الوثائق كطلبات عامّة للتعليقات (ومن ثم كان الاسم) لحل مشاكل تصميم الشبكة والبروتوكولات التي واجهت الشبكات الأولى التي سبقت الإنترنت. يغلب على تلك الوثائق الطبيعة الفنية والتفصيلية، حيث تقوم بتعريف البروتوكولات مثل TCP وIP وHTTP (للويب)، وSMTP (للبريد الإلكتروني). يوجد حالياً قرابة الـ 5000 طلب تعليقات. وتقوم هيئات أخرى أيضاً بوضع المعايير المتعلقة بمكونات الشبكة، وبشكلٍ خاص وصلات الشبكة. فعلى سبيل المثال تقوم لجنة IEEE 802 للشبكات المحلية (LANs) [IEEE 802 2007] بوضع معايير شبكات إيثرنت المحلية السلكية وشبكات WiFi المحلية اللاسلكية.

شبكة الإنترنت العامّة (أي الشبكة العالمية للشبكات التي نوقشت أعلاه) هي الشبكة التي نقصدها عادةً عندما نتحدث عن الإنترنت. هناك أيضاً العديد من الشبكات الخاصّة، كشبكات الشركات والجهات الحكومية، حيث لا تستطيع المضيفات فيها تبادل الرسائل مع المضيفات خارج الشبكة الخاصّة (ما لم تمر الرسائل عبر ما يسمّى ببرامج الحماية (firewalls) والتي تحد من تدفق الرسائل من الشبكة وإليها). تعرف تلك الشبكات الخاصة غالبا باسم شبكات الإنترانت (Intranet)، نظراً لكونها تستخدم نفس أنواع المضيفات، والموجّهات، والوصلات، والبروتوكولات المستخدمة على الإنترنت العامة.

2-1-1 وصف للخدمات

حددت المناقشة السابقة العديد من المكونات التي تشكّل الإنترنت. غير أنه يمكن أيضاً وصف الإنترنت من زاوية مختلفة تماماً – كبنية تحتية تزوّد التطبيقات بالخدمات المختلفة. تتضمّن تلك التطبيقات البريد الإلكتروني، وتصفح الويب، وإرسال الرسائل الفورية، ونقل الصوت باستخدام VoIP، وراديو الإنترنت، وعرض شرائط الفيديو، والألعاب الموزّعة، ومشاركة النظائر للملفات (P2P file sharing)، وتليفزيون الإنترنت، والاتصال عن بُعد، والكثير الكثير غير ذلك. تسمى التطبيقات تطبيقات موزّعة لأنها تتضمن عدة أنظمة طرفية تتبادل البيانات فيما بينها. ومن المهم ملاحظة أن تطبيقات الإنترنت يجري تشغيلها على الأنظمة الطرفية وليس على محوّلات الرزم في قلب الشبكة. كما سيتضح أكثر كلما مضينا قدُماً في هذا الكتاب، فرغم أن محوّلات الرزم تسهّل تبادل البيانات بين الأنظمة الطرفية، إلا أنها لا تُعنَى أبداً بالتطبيقات العاملة على الأنظمة الطرفية، والتي تمثل مصدر أو وجهة تلك البيانات.

لنستطرد قليلاً في توضيح مفهوم البنية التحتية التي توفر خدمات للتطبيقات. لنفترض أن لديك فكرة جديدة ومثيرة لتطبيق إنترنت موزّع، تطبيق قد يفيد الإنسانية كثيراً أو قد يجعلك ببساطة ثرياً ومشهوراً. ما الطريق الذي ستسلكه لتحويل تلك الفكرة إلى تطبيق إنترنت فعلي؟ حيث إن التطبيقات يتم تشغيلها على الأنظمة الطرفية فستحتاج لكتابة أجزاء برامج يتم تنفيذها على تلك الأنظمة. وقد تُقرر على سبيل المثال كتابة برامجك باستخدام لغة Java، أو C، أو C++. الآن ونظراً لأنك تطوّر تطبيقاً موزّعاً على الإنترنت، فإن أجزاء التطبيق التي تنفّذ على الأنظمة الطرفية المختلفة ستحتاج لإرسال البيانات إلى بعضها البعض. وهنا نصل إلى نقطة جوهرية – تلك التي تقودنا إلى الطريقة البديلة لوصف الإنترنت كمنصة للتطبيقات.

كيف يقوم جزء لتطبيق ينفّذ على نظام طرفي معين بتوجيه الإنترنت لتسليم البيانات إلى جزء آخر من التطبيق يجري تنفيذه على نظام طرفي آخر؟

توفر الأنظمة الطرفية الموصلة بالإنترنت واجهة برمجة التطبيقات (API) التي تحدد كيف يمكن لبرنامج ينفّذ على نظام طرفي أن يطلب من بنية الإنترنت التحتية تسليم البيانات إلى برنامجٍ مستهدفٍ بعينه يجري تنفيذه على نظام طرفي آخر. وتعرّف واجهة برمجة التطبيقات الخاصة بالإنترنت بمجموعة القواعد التي يتعين على البرامج المرسِلة اتباعها لكي تتمكن الإنترنت من تسليم البيانات إلى البرامج المستهدفة على الناحية الأخرى من الشبكة. وسوف نستعرض واجهة برمجة التطبيقات للإنترنت بالتفصيل في الفصل الثاني. سنكتفي الآن بالاستعانة بمثال تقريبي بسيط سنستخدمه كثيراً في هذا الكتاب. لنفترض أن أليس (Alice) تريد أن ترسل خطاباً إلى بوب (Bob) بالبريد. طبعاً لن يُعقل أن تقوم أليس بكتابة الرسالة (البيانات) ثم تلقي بها من نافذة غرفتها! بالتأكيد سيتطلّب نظام البريد أن تقوم أليس بوضع الرسالة في مظروف؛ وكتابة اسم بوب بالكامل، وعنوانه، والرمز البريدي في وسط المظروف؛ ثم إغلاق المظروف ووضع طابع البريد في الزاوية العليا اليمنى منه؛ وأخيراً وضع المظروف فى صندوق خدمة بريدية رسمي. من هنا يتضح أن لنظام البريد قواعد API خاصة به تتمثل في مجموعة الخطوات التي يتعين على أليس اتخاذها ليتم تسليم رسالتها إلى بوب. بالمثل يوجد للإنترنت قواعد API خاصة بها يتعين على قطعة برامج مرسِلة اتباعها لتتمكن الإنترنت من توصيل البيانات المطلوب نقلها إلى البرامج المستهدفة على مضيف آخر.

يوفر نظام البريد بالطبع عدة خدمات مختلفة لزبائنه، كالبريد المستعجل، وعِلم الوصول، والخدمة العادية، وغير ذلك من الخدمات. وكذلك توفر الإنترنت أيضاً خدمات متعدّدة لتطبيقاتها. لذا فعندما تقوم بتطوير تطبيق للإنترنت، عليك أيضاً أن تختار إحدى خدمات الإنترنت

لتطبيقك. وسوف نستعرض خدمات الإنترنت في الفصل الثاني ونتناول كيف تقوم الإنترنت بتوفير تلك الخدمات في الفصل الثالث.

يعتبر الوصف الثاني للإنترنت – كبنية تحتية لتوفير الخدمات اللازمة لتطبيقات موزّعة – مفهوماً مهماً، فالتقدّم في المكونات المادية للإنترنت تدفعه، بشكلٍ مضطرد، احتياجات التطبيقات الجديدة. لذا فمن المهم تذكّر أن الإنترنت بنية تحتية يجري تطوير واستخدام تطبيقات جديدة لها باستمرار.

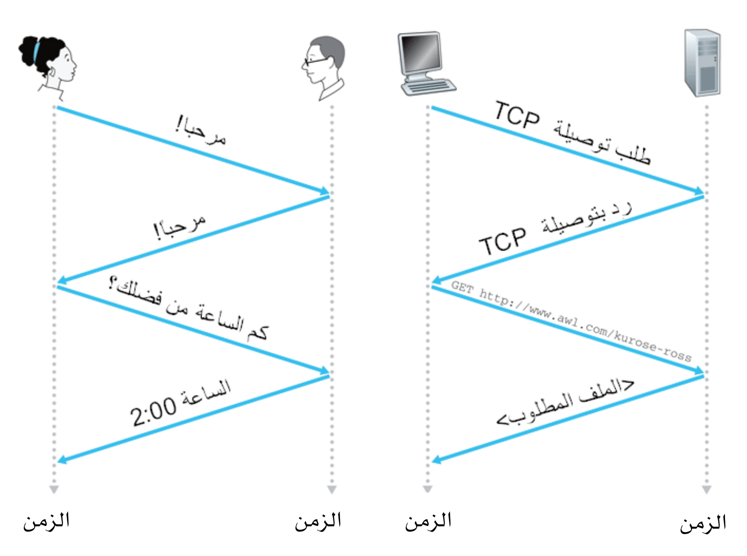
لقد أعطينا الآن وصفين للإنترنت، أولهما بدلالة مكوناتها المادية والبرمجية، والآخر من حيث كونها بنية تحتية لتوفير الخدمات للتطبيقات الموزّعة. لكن ربما لا تزال إجابة السؤال عن ماهية الإنترنت يشوبها بعض الغموض لديك. فما هو تحويل الرزم؟ وما هو نظام البروتوكولات TCP/IP؟ وما هي تعليمات API؟ وما هي الموجّهات؟ وما أنواع وصلات الاتصال الموجودة في الإنترنت؟ وماذا نعني بتطبيق موزّع؟ وكيف يمكن توصيل محمصة خبز (toaster) أو مجس (أداة استشعار) (sensor) لحالة الطقس بالإنترنت؟ إذا شعرت أن هذه التساؤلات تغمرك الآن فلا تقلق. فالغرض من هذا الكتاب تقديم وصف تفصيلي ودقيق لمكونات الإنترنت، بالإضافة إلى المبادئ التي تحكم كيف ولماذا تعمل. سنشرح تلك المصطلحات ونجيب عن تلك الأسئلة المهمة في الأجزاء والفصول التالية من الكتاب.

1-1-3 ما هو البروتوكول؟

الآن وقد أصبح لدينا فكرة عن ماهية الإنترنت، دعنا نأخذ بعين الاعتبار مصطلحاً مشهوراً آخر من مصطلحات شبكات الحاسب: "البروتوكول". فما هو البروتوكول؟ وما الدور الذي يلعبه؟ وكيف تُميِّز بروتوكولاً إن صادفته؟

مثال من واقع البشر:

قد يكون من الأسهل لفهم فكرة بروتوكولات شبكة الحاسب أن نأخذ بعين الاعتبار بعض الأمثلة من واقع البشر، ذلك أننا كبشر ننفّذ البروتوكولات طوال الوقت. تأمّل ما تفعله عندما تريد سؤال شخص ما عن الوقت. يبين الشكل 1-2 مثالاً للحديث المتبادل بهذا الصدد. يتطلب البروتوكول البشري (أو مبادئ السلوك الدمث على الأقل) أن يبدأ الطرف الأول بالتحية ("مرحباً" الأولى في الشكل 1-2) لبدء الاتصال مع شخص آخر. يأتي الردّ النمطي على "مرحباً" على شكل رسالة "مرحباً" أخرى. ضمنياً يَعتبر الشخص السائل الرد الودي بـ"مرحباً" كإشارة مشجعة على المضي قدُماً والسؤال عن الوقت. أي ردّ مختلف عن "مرحباً" (مثل "لا تزعجني!" أو "أنا لا أتحدث العربية") قد يشير إلى إحجامٍ أو عدم رغبة في الاتصال. في هذه الحالة وتبعاً للبروتوكول البشري لن يتم السؤال عن الوقت. أحياناً لا يتلقى السائل أي رد على الإطلاق، وفي هذه الحالة يتخلى عادةً عن المتابعة للسؤال عن الوقت. لاحظ أنه في مثالنا البشري، هناك رسائل معينة نرسلها، وأعمال معينة نقوم بها بناءً على الإجابة التي نتلقاها أو غير ذلك من الأحداث (مثل عدم تلقي إجابة في غضون مدة معينة). واضح أن إرسال واستقبال الرسائل وما يتبع ذلك من أعمال تشكل جزءاً محورياً في نظام البروتوكول البشري. أما إذا استخدم الناس بروتوكولات مختلفة (مثلاً إذا كان أحد الشخصين يتسم بسلوك جيد والآخر عكس ذلك، أو إذا كان مفهوم الوقت واضحاً عند طرف وغير واضح عند الآخر) فلن تؤدي البروتوكولات دورها ولن يتسنى إنجاز عمل مفيد. ينطبق الشيء نفسه على الشبكات، حيث يحتاج الأمر إلى اثنين أو أكثر من وحدات الاتصال تنفذ نفس البروتوكول لكي يتم إنجاز مهمّة معينة.



الشكل 1-2 بروتوكول بشري (إلى اليسار) وبروتوكول شبكة حاسب (إلى اليمين).

دعنا نتناول مثالاً بشرياً آخر. لنفترض أنك في فصل دراسي في إحدى الكليات (في حصة عن شبكات الحاسب مثلاً!)، والمعلّم يدندن حول موضوع البروتوكولات، والأمر مختلط عليك. ثم يتوقّف المعلّم ليسأل "هل هناك أي أسئلة؟" (رسالة ترسل إلى كلّ الطلاب، وتُستلم من قبَل أولئك غير النائمين في الفصل). تقوم أنت برفع يدك (ومن ثم ترسل رسالة ضمنية للمعلم). يبين المعلم تلقيه إشارتك بابتسامة قائلاً: "نعم". (رسالة تشجّعك على طرح سؤالك؛ فالمعلمون يحبون تلقي الأسئلة!)، تقوم أنت بعدها بإلقاء سؤالك (أي إرسال رسالتك إلى المعلّم). يسمع المعلّم سؤالك (يستلم رسالتك) ويجيبك عنها (يرسل الإجابة إليك). مرة أخرى نرى أنّ إرسال واستلام الرسائل ومجموعة الخطوات الروتينية التي تتخذ في تلك الأثناء تعتبر بمثابة القلب من هذا البروتوكول للسؤال والجواب.

تاريخ حالة (Case History)

تشكيلة مبهرة من أنظمة الإنترنت الطرفية:

منذ عهد ليس بالبعيد كانت أغلب النظم الطرفية الموصلة بالإنترنت حاسبات تقليدية كحاسبات المكتب والخادمات القوية. ومنذ أواخر التسعينيات وحتى اليوم يجري توصيل تشكيلة واسعة من الأجهزة تزداد تنوعاً وإثارة بالإنترنت. تلتقي هذه الأجهزة في خاصية مشتركة، ألا وهي احتياجها لتبادل البيانات الرقمية مع الأجهزة الأخرى. ونظراً للانتشار الواسع للإنترنت، ووضوح معالم بروتوكولاتها المعيارية، وتوافر أجهزة تجارية جاهزة للإنترنت، كان من الطبيعي استخدام تقنية الإنترنت لتوصيل تلك الأجهزة ببعضها.

تبدو بعض هذه الأجهزة كما لو كانت قد صممت للتسلية المحضة. فمثلاً يقوم إطار (برواز) صورة متوائم مع الإنترنت بتنزيل صور رقمية من خادم بعيد لعرضها على جهاز يشبه إطار صور تقليدي [Ceiva 2007]. كذلك قد تقوم محمصة الخبز المتصلة بالإنترنت بتنزيل المعلومات الإرصادية من خادم لتطبع صورة للطقس المتوقع اليوم (مثلاً غائم ومشمس جزئياً) على خبز الصباح [BBC 2001]. هناك أجهزة أخرى توفر معلومات مفيدة مثل كاميرات الويب التي تعرض حالة حركة المرور وأحوال الطقس، أو تراقب موقعاً محط اهتمام، وكذلك الأجهزة المنزلية المرتبطة بالإنترنت كالغسّالات والثلاجات والمواقد التى يمكن التحكم فيها عن بعد من خلال متصفحات الويب [Internet Home Alliance 2007]. كما تقوم الهواتف الخلوية المرتبطة بالإنترنت بوضع صفحات الويب ورسائل البريد الإلكتروني والرسائل الفورية عند أطراف أصابعك.

هناك صنف جديد من أنظمة المجسّات المشبّكة، يَعِد بثورة في الطريقة التي نراقب فيها بيئتنا ونتفاعل معها. حيث تسمح شبكات المجسات المثبتة فى البيئة الطبيعية بمراقبة المبانى والجسور والنشاط الزلزالي والحياة البرية ومصبّات الأنهار وطبقات الجو الدنيا [CENS 2007; Culler 2004; CASA 2007]. كما يمكن أيضاً دمج وتشبيك المجسات الطبية الحيوية [Schwiebert 2001]. ويتم توفير البيانات التي تجمعها المجسات للمستخدمين عن بُعدٍ في الوقت الحقيقي. فبوسع بطاقة الهوية اللاسلكية (RFID) أو مجس صغير مدمج مثبّت إلى أي شيء توفير المعلومات عنه على الإنترنت، مما سيؤدي في المستقبل إلى "إنترنت الأشياء" [ITU 2005].

بروتوكولات الشبكة

يشبه بروتوكول الشبكة البروتوكول المستخدم بين البشر (كما في الأمثلة السابقة)، مع الفارق أن الكيانات التي تتبادل الرسائل وتتخذ الخطوات هي مكونات مادية أو برمجية لبعض الأجهزة الموصلة بالشبكة (على سبيل المثال جهاز الحاسب أو المساعد الشخصي الرقمي أو الهاتف الخلوي أو الموجّهات أو غير ذلك من الأجهزة القادرة على التعامل مع الشبكة). كلّ أنشطة شبكة الإنترنت التي تتضمّن اثنين أو أكثر من الكيانات المتصلة عن بعد يحكمها بروتوكول. على سبيل المثال تقوم البروتوكولات المبنية في بطاقتى التوصيل بالشبكة ضمن جهازي حاسب موصلين مادياً بالتحكم في تدفق البتات على "السلك" الواصل بينهما، وتقوم بروتوكولات السيطرة على الازدحام الموجودة على الأنظمة الطرفية بتحديد معدل إرسال رزم البيانات من المرسل إلى المستقبل، كما تقوم بروتوكولات التوجيه في الموجّهات داخل الشبكة بتحديد المسار الذي تسلكه كل رزمة من المصدر إلى الوجهة النهائية. فالبروتوكولات موجودة في كل مكان في الإنترنت، ولذا فإن جزءاً كبيراً من هذا الكتاب ينصب على بروتوكولات شبكات الحاسب. وكمثال لبروتوكول شبكة حاسب، والذى قد يكون مألوفاً لديك، تأمل ما يحدث عندما تتقدم بطلب إلى خادم ويب (web server)، أي عندما تكتب عنوان (URL) لموقع صفحة ويب على متصفح الويب لديك. يبين النصف الأيمن من الشكل 1-2 ذلك السيناريو. في البداية يقوم جهاز حاسبك بإرسال طلب اتّصال إلى خادم الويب وينتظر إجابة. سيتسلم خادم الويب طلب اتّصالك في النهاية ويجيب برسالة قبول الاتصال. عندئذٍ يدرك حاسبك أنه من المناسب الآن طلب صفحة الويب التي يريد جلبها من الخادم، فيرسل اسم الصفحة ضمن رسالة طلب إحضار GET، وأخيراً يرسل خادم الويب صفحة الويب المطلوبة إلى حاسبك.

من خلال الأمثلة البشرية وأمثلة الشبكات التي استعرضناها أعلاه، يتضح أن عملية تبادل الرسائل والخطوات التي تصاحب إرسال واستقبال تلك الرسائل تمثل العناصر الرئيسية للبروتوكول:

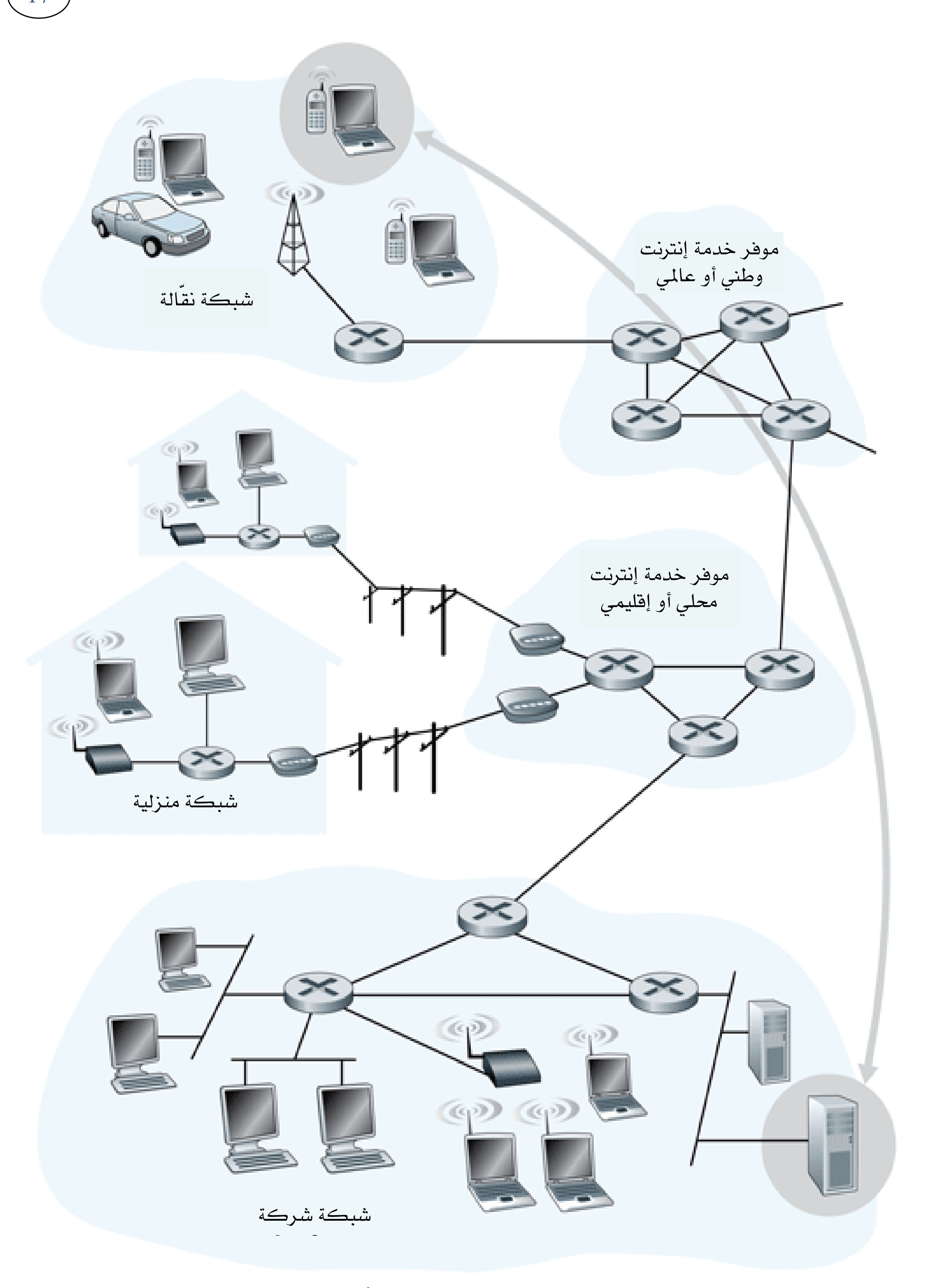
"يُعرِّف البروتوكول صيغ وترتيب الرسائل التي يتم تبادلها بين اثنين أو أكثر من الكيانات المتصلة، بالإضافة إلى الخطوات التي تُتخذ عند إرسال أو استقبال رسالة أو حصول حدث آخر."

تستخدم الإنترنت – وشبكات الحاسب عموماً – البروتوكولات بشكلٍ مكثف، حيث يستخدم العديد من البروتوكولات في إنجاز مهام الاتصالات المختلفة. من خلال قراءتك لهذا الكتاب، ستعلم أن بعض البروتوكولات بسيطة ومباشرة، بينما البعض الآخر معقّد ويتسم بالعمق. يتطلب إتقان مجال شبكات الحاسب فهم "ماذا" و"لماذا" و"كيف" المتعلقة ببروتوكولات الشبكات.

2-1 حافة الشبكة

قدّمنا في الجزء السابق من هذا الفصل نظرة عامّة عن بروتوكولات الشبكات والإنترنت. سنغوص الآن بشكلٍ أعمق للتعرف على مكونات شبكة حاسب (وشبكة الإنترنت بشكلٍ خاص). سنبدأ فى هذا الجزء عند حافة الشبكة لننظر إلى المكونات المألوفة جداً لدينا: كالحاسبات، والمساعدات الرقمية الشخصية، والهواتف الخلوية، وغير ذلك من الأجهزة الأخرى التي نستعملها بشكلٍ يومي. في الجزء التالي سننتقل من حافة الشبكة إلى قلبها ونستعرض عمليات التحويل والتوجيه في شبكات الحاسب.

وجدنا في الجزء السابق من هذا الفصل أنه تبعاً لمفردات شبكات الحاسب، غالباً ما يطلق على الحاسبات والأجهزة الأخرى الموصلة بشبكة الإنترنت الأنظمة الطرفية. يطلق عليها هذا الاسم لأنها توجد على حافة الإنترنت، كما هو مبين في الشكل 1-3. وتتضمّن أنظمة الإنترنت الطرفية حاسبات المكتب (كحاسبات المكتب الشخصية، وأجهزة الماكنتوش، وحاسبات لاينكس)، والخادمات (كخادمات الويب والبريد الإلكتروني)، والحاسبات النقّالة (كالحاسبات النقالة، والمساعدات الرقمية الشخصية، والهواتف الموصّلة لاسلكياً بالإنترنت). وعلاوة على ذلك هناك عدد متزايد من الأجهزة الأخرى ترتبط الآن بالإنترنت كأنظمة طرفية (انظر شريط المعلومات الثانوى بالصفحة التالية).



الشكل 1-3 التفاعل فيما بين الأنظمة الطرفية.

تُعرف الأنظمة الطرفية أيضاً باسم "المضيفات" لأنها تستضيف (أي تقوم بتشغيل) برامج تطبيقية كبرنامج متصفح الويب، وبرنامج خادم الويب، وبرنامج قارئ البريد الإلكتروني، وبرنامج خادم البريد الإلكتروني. في هذا الكتاب سنستخدم المصطلحين "مضيف" و"نظام طرفي" بنفس المعنى (أي أن مضيف = نظام طرفي). أحياناً تُقسّم المضيفات أبعد من ذلك إلى فئتين: الزبائن (clients) والخادمات (servers). وبصفة عامة، يغلب أن يكون "الزبون" حاسباً مكتبياً، أو حاسباً شخصياً نقّالاً، أو مساعداً رقمياً شخصياً وهلم جرا. بينما غالباً ما يكون "الخادم" جهاز حاسب أكثر قوة لتخزين وتوزيع صفحات الويب، وعرض شرائط الفيديو، وتحويل رسائل البريد الإلكتروني، وما شابه ذلك.

1-2-1 برامج الزبائن والخادمات

من منظور برمجيات الشبكة، هناك تعريف آخر للزبون والخادم وهو ما سنستخدمه في هذا الكتاب. برنامج الزبون هو عبارة عن برنامج يُنفّذ على نظام طرفي يقوم بطلب وتلقّي خدمة من برنامج خادم ينفّذ على نظام طرفي آخر. ويعتبر نموذج الزبون – الخادم هذا بلا شك النموذج الأكثر انتشاراً في تطبيقات الإنترنت. حيث تستخدم هذا النموذج تطبيقات الويب والبريد الإلكتروني ونقل الملفات والدخول على الحاسبات عن بُعدٍ (على سبيل المثال Telnet)، ومجموعات تبادل الأخبار، والعديد من التطبيقات الشهيرة الأخرى. ولمّا كان برنامج الزبون ينفّذ على حاسب بينما ينفّذ برنامج الخادم على حاسب آخر، فإن تطبيقات الإنترنت من نوع الزبون – الخادم، تعتبر تطبيقات موزّعة من حيث التعريف. يتفاعل برنامج الزبون وبرنامج الخادم عن طريق تبادل الرسائل فيما بينهما عبر الإنترنت. في هذا المستوى من التصوير التجريدى، تعمل الموجّهات والوصلات وغيرها من المكونات التفصيلية للإنترنت بشكلٍ جماعي كصندوق أسود مهمته نقل الرسائل بين المكونات الموزعة لتطبيق على الإنترنت، كما هو موضح فى الشكل 1-3.

يلاحظ أن تطبيقات الإنترنت الآن لا تتكون كلها من برامج زبائن خالصة تتفاعل مع برامج خادمات خالصة. فعلى نحو متزايد ظهرت العديد من التطبيقات

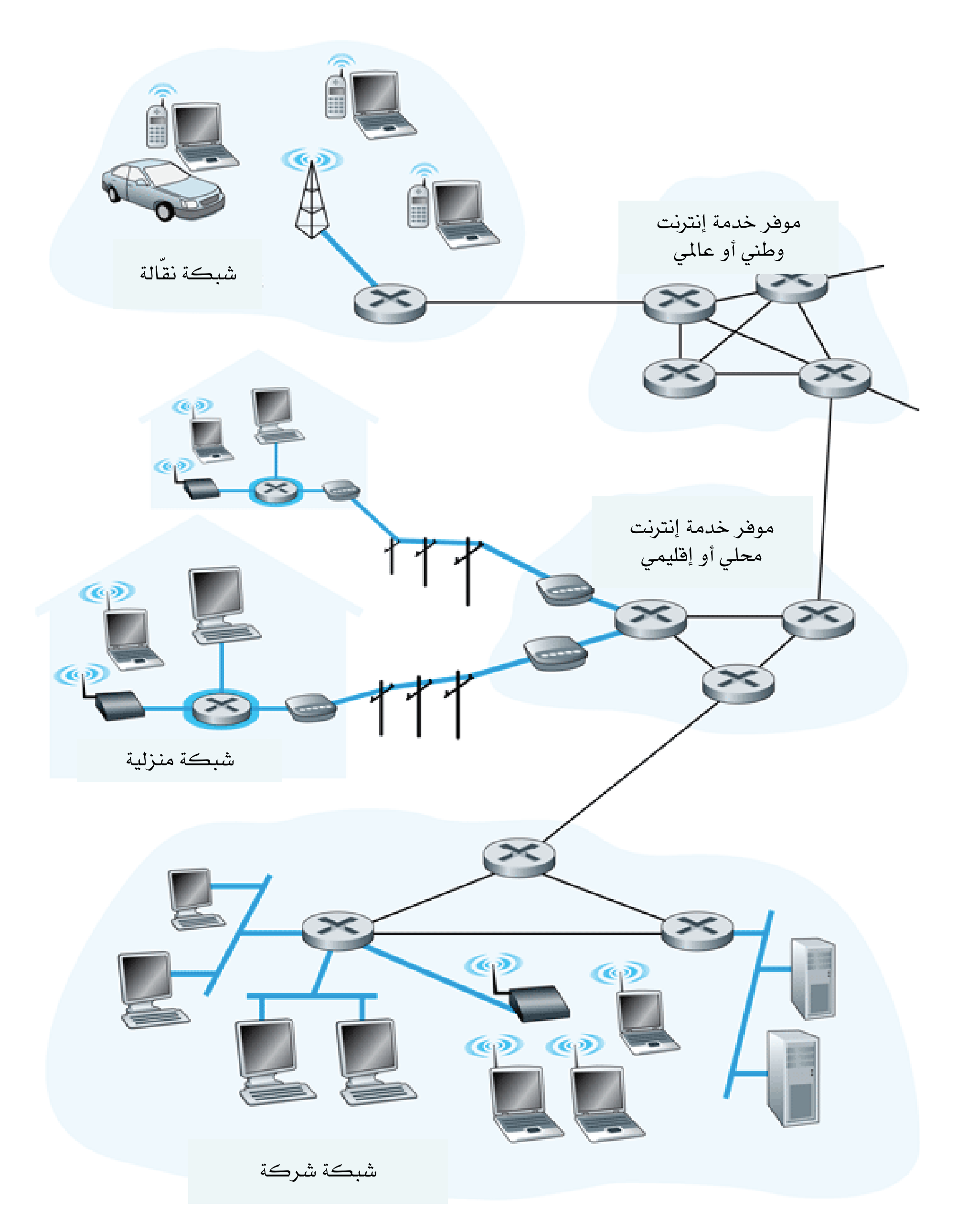
اليوم من نوع تطبيقات النظائر P2P، حيث تتفاعل الأنظمة الطرفية وتقوم بتشغيل برامج تؤدّي وظائف كلٍّ من الخادم والزبون. على سبيل المثال في تطبيقات النظائر لمشاركة الملفات (مثل LimeWire، وeDonkey، وKazaa) يتصرف البرنامج في النظام الطرفي لمستخدم ما كزبون عندما يطلب ملفاً من نظير آخر، وكخادم عندما يرسل ملفاً إلى نظير آخر. وفي نظام هاتف الإنترنت يتفاعل طرفا الاتصال كنظيرين ضمن جلسة اتصال متماثلة يقوم فيها الطرفان بإرسال واستلام البيانات. وسنقوم بمقارنة أوجه الشبه والاختلاف بين نموذج زبون/خادم ونموذج النظائر في الفصل الثاني.

2-2-1 شبكات الوصول (Access Networks)

بعد أن استعرضنا التطبيقات والأنظمة الطرفية عند "حافة الشبكة"، دعنا نتناول موضوع شبكات الوصول – أي تلك الوصلة أو الوصلات المادية التي تربط نظاماً طرفياً بموجّه الحافة الخاص به (وهو الموجّه الأول على المسار من ذلك النظام الطرفي إلى أي نظام طرفي آخر بعيد). يبين الشكل 1-4 عدّة أنواع من الوصلات التي تستخدم لربط نظام طرفي بموجّه الحافة، حيث تبين وصلات الوصول تلك في الشكل بخطوط مظلّلة سميكة. يمكن تصنيف شبكات الوصول على وجه التقريب إلى ثلاث فئات:

- شبكات الوصول السكني: لربط الأنظمة الطرفية المنزلية بالشبكة.
- شبكات الوصول التجاري: لربط الأنظمة الطرفية في شركة تجارية أو مؤسسة تعليمية بالشبكة.
- شبكات الوصول اللاسلكي: لربط الأنظمة الطرفية (والنقالة في أغلب الأحيان) بالشبكة لاسلكياً.

هذا التقسيم ليس محدداً بشكلٍ صارم. فعلى سبيل المثال قد تستخدم بعض الأنظمة الطرفية في شركةٍ ما تقنية وصول تنسب بشكلٍ عام للوصول السكني، والعكس صحيح. إلا أن الأوصاف المذكورة تنطبق على الحالات العامة.



الشكل 1-4 شبكات الوصول.

شبكات الوصول السكني

يُقصد بالوصول السكني توصيل نظام طرفي منزلي (حاسب شخصي أو شبكة منزلية) إلى موجّه حافة. يستخدم أحد أشكال الوصول السكني المودم الهاتفي (dial-up modem) على خطّ هاتف عادي للربط بموفر خدمة إنترنت سكني (مثل أميريكا أون لاين AOL). يُحوّل المودم المنزلي البيانات الرقمية الخارجة من الحاسب الشخصي إلى صيغة تناظرية (analog) صالحة للإرسال على خطّ الهاتف، والذى يتكون من زوج مجدول من الأسلاك النحاسية (سنتناول زوج الأسلاك النحاسية المجدول بالتفصيل لاحقاً في هذا الجزء). على الطرف الآخر لخطّ الهاتف التناظري يقوم مودم موفر خدمة الإنترنت بتحويل الإشارة التناظرية المستقبلة إلى بيانات رقمية لإدخالها لموجّه موفر الخدمة. وهكذا فإن شبكة الوصول في هذه الحالة تتكون ببساطة من زوج من المودمات بينهما خط هاتف. تسمح سرعات أجهزة المودم المتوفرة اليوم بنقل البيانات بهذه الطريقة بسرعات تصل إلى 56 كيلوبت/ثانية. ومع ذلك، فغالباً ما يحصل الكثير من مستخدمي هذه الطريقة على سرعات فعّالة تقل بكثير عن 56 كيلوبت/ثانية بسبب النوعية الرديئة لخطوط أزواج الأسلاك المجدولة التي تصل العديد من المنازل بموفري خدمة الإنترنت.

يجد العديد من المستخدمين السكنيين سرعة الوصول 56 كيلوبت/ثانية بطيئة بشكلٍ لا يطاق. فعلى سبيل المثال، يستغرق تنزيل مادة مسموعة بصيغة MP3 مدتها ثلاث دقائق على مودم هاتفي سرعته 56 كيلوبت/ثانية حوالي ثماني دقائق. وعلاوة على ذلك يشغل استخدام المودم الهاتفي لتصفّح الويب خط الهاتف العادي، فلا يستطيع المستخدم أثناء ذلك تلقي أو عمل مكالمات هاتفية على نفس الخط. ولحسن الحظ توفر تقنيات الوصول الحديثة بحيز ترددي عريض للمستخدمين السكنيين سرعات عالية لنقل البيانات ووسيلة لاستخدام الإنترنت مع إمكانية استخدام الهاتف للمكالمات في نفس الوقت. توجد طريقتان رئيستان للوصول السكني بحيز ترددي عريض: خط

المشترك الرقمي DSL [DSL 2007]، والشبكات الهجينة (Hybrid Fiber-Coaxial Cable (HFC)) (والتي تستخدم الكبلات المحورية مع كبلات الألياف الضوئية) [Cable Labs 2007].

في مارس/آذار عام 2006 كانت الخطوط بحيز ترددي عريض تتوافر فيما يزيد على 50٪ من المنازل في العديد من البلدان المتقدمة، وبمعدل أعلى من 80٪ في كوريا الجنوبية وهونج كونج. تتقدم كلٌّ من الولايات المتحدة والصين بقية العالم من حيث العدد الكلي لتلك الخطوط، والذي يصل إلى 40 مليون خط تقريباً في كلٍّ منهما [Point Topic 2006]. وفي ذلك الوقت كان حوالي 60٪ من تلك الخطوط في الولايات المتحدة وكندا من نوع HFC، بينما بقية الخطوط من نوع DSL. أما خارج الولايات المتّحدة وكندا، فينتشر استخدام DSL، خصوصاً في أوروبا حيث تتجاوز نسبته في العديد من بلدانها 90٪.

عادةً ما تقوم شركات الهاتف (على سبيل المثال Verizon أو France Telecom) بتوفير خدمة DSL للوصول السكني مباشرةً أو بالاشتراك مع موفر خدمة إنترنت مستقل. تشبه تقنية DSL من حيث المبدأ تقنية المودم الهاتفي، فهي تستخدم نفس خطوط الهاتف السلكية المجدولة الموجودة حالياً. فبوضع حدٍّ أقصى للمسافة بين المستخدم وموفر خدمة الإنترنت يمكن بهذه الطريقة إرسال واستلام البيانات بمعدلات أعلى بكثير. وعادةً ما تكون معدلات نقل البيانات غير متماثلة في الاتجاهين، حيث يكون المعدل من موجّه موفر الخدمة إلى المنزل أعلى منه من المنزل إلى موفر الخدمة. يعكس عدم التماثل هذا الاعتقاد بأن المستخدم المنزلي يغلب عليه على الأرجح أن يكون مستهلكاً للمعلومات (أى يجلب البيانات إلى منزله) أكثر من كونه منتجاً لها.

تقسّم تقنية DSL وصلة الاتصال بين المنزل وموفر خدمة الإنترنت إلى ثلاثة نطاقات تردد مستقلة:

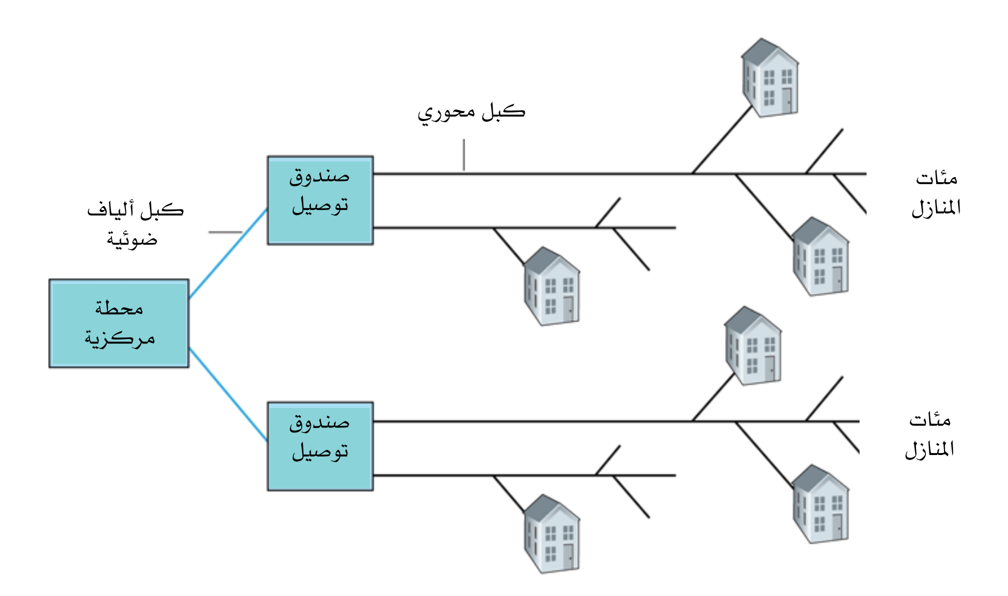
- قناة عالية السرعة على الحيز الترددي (50 كيلوهرتز – 1 ميجاهرتز) من موفر الخدمة إلى المنزل (ويسمى بالاتجاه النازل من الإنترنت downstream).

- قناة متوسطة السرعة على الحيز الترددي (4 كيلوهرتز – 50 كيلوهرتز) من المنزل إلى موفر الخدمة (ويسمى بالاتجاه الصاعد إلى الإنترنت upstream).
- قناة هاتف عادية مزدوجة الاتجـــاه على الحيز التـــرددي (صفر كيلوهرتز – 4 كيلوهرتز).

بهذه الطريقة تبدو وصلة DSL الواحدة كما لو كانت ثلاث وصلات مستقلة حيث يمكن لمكالمة هاتفية واتصال إنترنت في الاتجاهين بالانتقال على وصلة DSL في نفس الوقت (سنتناول هذه التقنية للإرسال المتعدد بتقسيم التردد في الجزء 1-3-1). وجدير بالذكر أن المعدلات الفعلية لإرسال البيانات في كلا الاتجاهين تعتمد على المسافة بين مودم المنزل ومودم موفر خدمة الإنترنت، ونوع ومقاس زوج الأسلاك المجدولة، ودرجة التداخل الكهرومغناطيسي، وغير ذلك من الاعتبارات الأخرى. واضح أن المهندسين صمّموا تقنية DSL للمسافات القصيرة بين المودمات السكنية ومودمات موفر خدمة الإنترنت (خلافاً لمودمات الهاتف العادية)، ولذا فهي تسمح بمعدلات أعلى بكثير لإرسال البيانات مقارنةً بطريقة الإرسال التقليدية عبر خط الهاتف. تتوافر عادةً معدلات إرسال مختلفة بأسعار مختلفة للمستخدمين القريبين من مودم موفر خدمة الإنترنت. فعلى سبيل المثال توفر شركة Verizon معدلات إرسال تقريبية تتراوح مابين 768 كيلوبت/ثانية و1.5 ميجابت/ثانية إلى المنزل، وما بين 128 كيلوبت/ثانية إلى 768 كيلوبت/ثانية من المنزل. وقد ظهرت أيضاً تشكيلة حديثة من تقنيات DSL بسرعات أعلى وحققت مؤخراً انتشاراً سريعاً في عدد من البلدان. فعلى سبيل المثال توفر تقنية DSL عالية السرعة (VDSL) المنتشرة بكثرة في كوريا الجنوبية واليابان الآن معدلات إرسال رائعة تبلغ 12-55 ميجابت/ثانية إلى المنزل و1.6-20 ميجابت/ثانية من المنزل [DSL 2007].

بينما تستخدم تقنية DSL والمودمات الهاتفية خطوط الهاتف العادية، تعتبر شبكات الوصول الهجينة (HFC) امتداداً لشبكات الكبل الحالية

المستخدمة لبث إشارات تليفزيون الكبل (Cable TV). في نظام تليفزيون الكبل التقليدي، تقوم محطة مركزية بالبث إلى المنازل عبر شبكة توزيع تضم كبلات محورية ومضخمات (amplifiers). وكما يوضح الشكل 1-5، تستخدم ألياف ضوئية لتوصيل المحطة المركزية إلى صناديق اتصال على مستوى الحي والتي تنطلق منها كبلات محورية تقليدية لتصل إلى منازل وشقق المشتركين، ويقوم كل صندوق بالحي بتغذية عددٍ من المنازل يتراوح عادةً ما بين 500 إلى 5000 منزل.



الشكل 1-5 شبكة وصول هجينة من كبلات الألياف الضوئية والكبلات المحورية.

كما هو الحال مع تقنية DSL، تتطلب تقنية HFC أجهزة مودم خاصة من نوع مودم الكبل، والذي يتم الحصول عليه من شركة توفير الوصول للإنترنت إما بالشراء أو الاستئجار. يُعتبر مودم الكبل عادةً أداة خارجية ويوصل إلى الكمبيوتر الشخصي بالمنزل عبر منفذ إيثرنت (سنتناول الإيثرنت بالتفصيل في الفصل الخامس). وتقسّم أجهزة مودم الكبل شبكة الـ HFC إلى قناتين: واحدة إلى المنزل (downstream) والأخرى من المنزل (upstream).

وعادةً ما تخصص للقناة التي تنقل البيانات إلى المنزل معدلات إرسال أعلى من القناة الأخرى كما هو الحال في تقنية DSL.

من الخواص المهمة لتقنية HFC استخدامها لوسط بث مشترك بحيز ترددي عريض. وبالتحديد، فإن كل رزمة بيانات تبث من المحطة المركزية تنتقل بالفعل على كل وصلة وتصل إلى كل منزل؛ كما أن كل رزمة ترسل من أي منزل تنتقل عبر القناة من المنزل إلى المحطة المركزية. ولهذا السبب فعندما يقوم عدّة مستخدمين بتنزيل ملفات MP3 في نفس الوقت على القناة من المحطة المركزية إلى المنازل، فإن المعدل الفعلي لنقل بيانات MP3 لكل مستخدم سيكون أقل بكثير من المعدل الكلي فى الاتجاه النازل (أى من المحطة المركزية إلى المنازل). من جانب آخر إذا كان هناك فقط بضعة مستخدمين نشطين وكلهم منهمكون في تصفح الويب، ففي الواقع قد يتلقى كل منهم صفحات الويب الخاصة به بالمعدل الكلى فى الاتجاه النازل، ذلك لأنه نادراً ما يقوم المستخدمون بطلب صفحة ويب فى نفس الوقت بالضبط.

ونظراً لأن القناة الصاعدة إلى المحطة المركزية مشتركة أيضاً، فإن الأمر يحتاج إلى بروتوكول موزّع للتحكم فى الوصول المتعدد بتنظيم توقيت الإرسال وتجنب التصادمات (collisions) (سنناقش قضية التصادمات بشيء من التفصيل عند تناولنا لشبكة الإيثرنت في الفصل الخامس). لهذا قد يشير المتحمسون لتقنية DSL بسرعة إلى أن تلك التقنية تعتمد على وصلة من نقطة إلى نقطة بين المنزل وموفر خدمة الإنترنت، ولذا فإن سعة إرسال وصلة DSL بكاملها تكون مخصصة لمستخدم واحد وليست مشتركة بين عدد من المستخدمين. أما أنصار الكبل فيشيرون إلى أن شبكة HFC بحجم معقول توفر معدلات إرسال أعلى من نظام DSL. إن المعركة بين تقنيتي DSL وHFC لكسب سوق الوصول السكني عالي السرعة محتدمة الآن، وخاصة في أمريكا الشمالية. في المناطق الريفية النائية ـ حيث لا تتوافر أي من تقنتي DSL أو HFC – يمكن استخدام وصلة قمر صناعي لتوصيل مسكن إلى الإنترنت بسرعة تتجاوز 1 ميجابت/ثانية.

ومن بين موفري خدمة الإنترنت بالقمر الصناعي شركات مثل StarBand وHughesNet.

من الميزّات الجذابة لتقنيات DSL وHFC والوصول بالقمر الصناعي أن الخدمة تكون متوفرة باستمرار (always-on)، فبوسع المستخدم ترك حاسبه شغالاً طوال الوقت ليبقى موصلاً بموفر خدمة الإنترنت بشكلٍ دائم بينما يقوم بعمل وتلقي المكالمات الهاتفية العادية بنفس الوقت.

شبكات الوصول التجاري

فى مقار الشركات والمدن الجامعية تستخدم شبكة محلية (LAN) عادةً لتوصيل نظام طرفي إلى موجّه الحافة. وكما سنرى في الفصل الخامس هناك العديد من تقنيات شبكات الاتصال المحلية (LAN)، غير أن تقنية الإيثرنت تعتبر حالياً أكثر تقنيات الوصول انتشاراً فيما يتعلق بشبكات الشركات. تعمل تقنية الإيثرنت اليوم بسرعات 100 ميجابت/ثانية أو 1 جيجابت/ثانية (أو حتى 10 جيجابت/ثانية)، وتستخدم إما زوج أسلاك نحاسية مجدولة أو كبلاً محورياً لتوصيل عددٍ من الأنظمة الطرفية مع بعضها البعض وبموجّه حافة. يضطلع موجّه الحافة بمسؤولية توجيه رزم البيانات التي تكون وجهتها خارج الشبكة المحلية. تستخدم الإيثرنت وسط توصيل مشترك ـ مثلها في ذلك مثل تقنية HFC — ومن ثم فإن المستخدمين يشتركون في معدل إرسال البيانات على الشبكة المحلية. وقد شهدت الفترة الأخيرة انتقالاً من تقنية الإيثرنت المشتركة (shared) إلى تقنية الإيثرنت المحوّلة (switched). وتستخدم تقنية الإيثرنت المحوّلة الشكل النجمي للشبكة (star topology)، حيث تُوصل المضيفات مباشرة إلى محوّل (switch)، مما يسمح لكل المضيفات بإرسال واستلام البيانات في نفس الوقت عند معدل الإرسال الكامل للشبكة المحلية. وسوف نتناول كلاً من الإيثرنت المشتركة والمحوّلة بالتفصيل في الفصل الخامس.

شبكات الوصول اللاسلكي

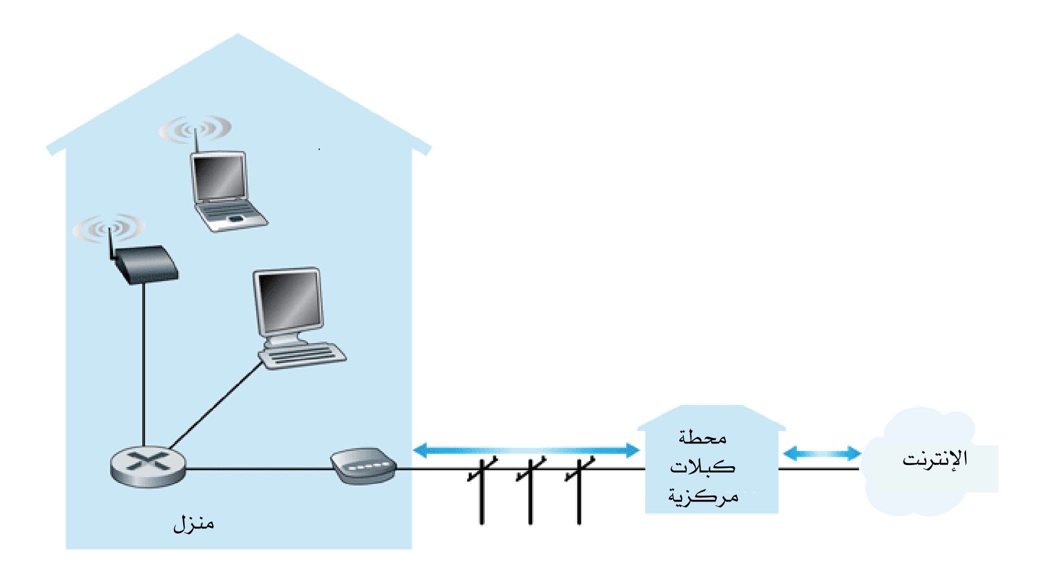
تتزامن مع ثورة الإنترنت الحالية ثورة موازية في تقنية الاتصال اللاسلكي والتي تلعب هي الأخرى دوراً كبيراً في الكيفية التي يعمل بها الناس ويعيشون. فاليوم في أوربا يفوق عدد الناس الذين لديهم هاتف جوال عدد الذين لديهم حاسب شخصي أو سيارة. والاتجاه لاستخدام الوصول اللاسلكي مستمر باضطراد، حيث يتوقع كثير من المحللين أن تصبح الأجهزة اللاسلكية النقّالة والمحمولة باليد في أغلب الأحيان، كالهواتف الخلوية والمساعدات الرقمية الشخصية الوسيلة الرئيسية للوصول إلى الإنترنت في كافة أنحاء العالم بدلاً من الحاسبات المشبّكة سلكياً. ينتشر حالياً نوعان من طرق الوصول اللاسلكي للإنترنت. يتمثل النوع الأول في الشبكات المحلية اللاسلكية (WLAN) حيث يقوم مستخدمو اللاسلكي بإرسال الرزم إلى محطة القاعدة (base station) (والتي تُعرف أيضاً بنقطة الوصول اللاسلكي) واستلامها منها على مدى نصف قطر يبلغ بضع عشرات من الأمتار. عادةً ما توصل محطة القاعدة بالإنترنت بواسطة أسلاك، ومن ثم فهي توصّل مستخدمي اللاسلكي إلى الشبكة السلكية. أما النوع الثاني فيستخدم شبكات وصول لاسلكي تغطى مساحات واسعة يتم عبرها إرسال الرزم على نفس البنية التحتية اللاسلكية الخاصة بشبكة الهاتف الخلوي، حيث تدار محطة القاعدة في هذه الحالة من قِبَل موفر خدمة الاتصالات السلكية واللاسلكية. توفر هذه الطريقة وصولاً لاسلكياً للمستخدمين ضمن نصف قطر يصل إلى عشرات الكيلومترات من محطة القاعدة.

تتمتع الشبكات المحلية اللاسلكية والمبنية على تقنية IEEE 802.11 (والتي تعرف كذلك بالإيثرنت اللاسلكية أو WiFi) بالانتشار الواسع حالياً في أقسام الجامعات، ومكاتب الشركات، والمقاهي، والمنازل. فالعديد من الجامعات تستخدم محطات مرجع IEEE 802.11 عبر الحرم الجامعي مما يسمح للطلاب بإرسال واستلام البريد الإلكتروني أو تصفّح الويب من أي مكان داخل الحرم الجامعي (مثل المكتبة، أو غرفة السكن، أو قاعة الدروس، أو مقعد في

الهواء الطلق). في العديد من مدن العالم الآن، بوسع الشخص أن يقف على ناصية شارع ويكون في نطاق عشر أو عشرين محطة مرجع. (راجع [wiggle.net 2007] للاطلاع على خريطة عالمية لمواقع محطات 802.11 المرجعية التي تم اكتشافها وتسجيلها على الويب). وتوفر تقنيات 802.11 الأكثر انتشاراً (والتي سنتناولها بالتفصيل في الفصل السادس) وسطاً مشتركاً للاتصال بمعدل إرسال يبلغ 54 ميجابت/ثانية.

يجمع العديد من المنازل اليوم ما بين وسائل الوصول السكني بحيز ترددي عريض (كمودم الكبل أو وصلة DSL) وتقنية الشبكات المحلية اللاسلكية الرخيصة (WLAN) للحصول على شبكات منزلية قوية. يبين الشكل 1-6 مخططاً لشبكة منزلية نمطية تتكون من حاسب محمول متنقل بالإضافة إلى حاسب شخصي مشبك سلكياً؛ ومحطة مرجع (نقطة وصول لاسلكي) توفر الاتصال بالحاسب النقال؛ ومودم كبل يوفر وصولاً عريض النطاق للإنترنت؛ وموجّه يربط محطة القاعدة والحاسب الشخصى الثابت بمودم الكبل. توفر هذه الشبكة لأهل المنزل وصولاً للإنترنت بحيز ترددي عريض، حيث يمكن لأحد الأفراد التجوّل بالحاسب النقال من المطبخ إلى الفناء الخلفي إلى غرف النوم. تبلغ التكلفة الثابت الكلية لمثل هذه الشبكة أقل من 150 دولاراً (بما في ذلك تكلفة مودم الكبل أو وصلة DSL).

للدخول على الإنترنت من خلال شبكة محلية لاسلكية، تحتاج عادةً لتكون في حدود بضع عشرات الأمتار من محطة القاعدة. وهذا الشرط ممكن عند الوصول من منزل، أو من مقهى، أو عموماً من داخل بناية أو حولها. لكن ماذا لو كنت على الشاطئ أو داخل سيارتك وتحتاج للاتصال بالإنترنت؟ للوصول عبر منطقة واسعة، يمكن لمستخدمي الإنترنت الجوّالين استعمال البنية التحتية للهاتف الخلوي، للوصول عبر محطات المرجع الموجودة داخل منطقة فى حدود عشرات الكيلومترات. من حيث المبدأ يشبه ذلك وصول شخص إلى الإنترنت عن طريق مودم وخط هاتف سلكي، مع فارق استخدام البنية التحتية للهواتف الخلوية في الإرسال بدلاً من بنية شبكة الهاتف السلكية.



الشكل 1-6 مخطط لشبكة منزلية نمطية.

لقد استثمرت شركات الاتصالات مبالغ طائلة في ما يسمّى بخدمات الجيل الثالث (3G Services) اللاسلكية، والتي توفر وصولاً لاسلكياً بالإنترنت بتحويل الرزم عبر منطقة واسعة بسرعات تزيد على 1 ميجابت/ثانية. يوجد معياران قياسيان هامان للإنترنت اللاسلكية عبر منطقة واسعة هما Evolution-(EVDO)) (Data Optimized و(High-Speed Downlink Packet Access (HSDPA)) حيث يوفر العديد من مشغلي شبكات الهاتف الخلوي ـ أو يخطّطون لتوفير ـ أحد هذين المعيارين. وكما هو الحال دائماً توجد تقنية مكتسحة ومرشحة لإسقاط هذين المعيارين من على عرشهما، ألا وهي تقنية WiMAX، والتي تُعرَف كذلك بـ IEEE 802.16، وتعتبر قريبة لبروتوكول WiFi (الذي ذكرناه أعلاه) ولكن للمسافات البعيدة [Intel WiMAX 2007; WiMAX Forum 2007]. تعمل تقنية WiMAX بشكلٍ مستقل عن الشبكة الخلوية وتعد بتحقيق سرعات تتراوح ما بين 5 إلى 10 ميجابت/ثانية أو أعلى عبر مسافات تصل إلى عشرات الكيلومترات. ورغم أنه لم يتحقق انتشار واسع لتقنية WiMAX حتى أواخر عام 2006، فقد خصصت

مجموعة Sprint-Nextel بلايين الدولارات لنشر WiMAX في عام 2007 وما بعده. وسنقوم بتغطية تقنيات WiFi ، وWiMAX ، و3G بالتفصيل في الفصل السادس.

1-2-3 أوساط الاتصال المادية

في الجزء السابق من هذا الفصل ألقينا نظرة عامّة على عددٍ من أهم تقنيات شبكات الوصول للإنترنت. من خلال وصفنا لتلك التقنيات أشرنا أيضاً إلى الأوساط المادية المستخدمة لتحقيق عملية الاتصال. فمثلاً ذكرنا أن تقنية HFC تستخدم خليطاً من الألياف الضوئية والكبلات المحورية، وقلنا: إنّ أجهزة المودم الهاتفية بسرعة 56 كيلوبت/ثانية وتقنية DSL تستخدم زوجاً من الأسلاك النحاسية المجدولة، كما ذكرنا أن شبكات الوصول النقّالة تستخدم طيف ترددات الراديو. في هذا الجزء سنقدم نظرة عامّة مختصرة عن تلك الأوساط وغيرها من أوساط الإرسال المستخدمة بكثرة في الإنترنت.

لكي نُعرّف ما الذي نعنيه بالتحديد بوسط مادي، دعنا نتأمل الحياة القصيرة لبت بيانات واحد. لنأخذ بتاً ينتقل من نظام طرفي عبر سلسلة من الوصلات والموجّهات إلى نظام طرفي آخر. يتم دفع ذلك البت وإرساله مرات عديدة من وصلة إلى وصلة! يبدأ النظام الطرفي المصدر بإرسال البت أولاً، وبعد ذلك بقليل يستقبله الموجّه الأول في السلسلة ثمّ يرسله، ليستلمه بعد ذلك بقليل الموجّه الثاني ويرسله وهكذا دواليك. يتضح من ذلك أن ذلك البت يمر أثناء انتقاله من المصدر إلى وجهته النهائية عبر سلسلة أزواج من المرسلات/المستقبلات حيث يتم إرساله بين كل مرسل ومستقبل عن طريق انتقال الموجات الكهرومغناطيسية أو النبضات الضوئية عبر وسط مادي. يمكن أن يأخذ الوسط المادي العديد من الأشكال والصور ولا يحتاج بالضرورة أن يكون من نفس النوع بين كل أزواج المرسلات والمستقبلات على طول المسار. تتضمّن أنواع الأوساط المادية أزواج أسلاك النحاس المجدولة، والكبلات المحورية، وكبلات الألياف الضوئية متعدّدة الأنماط، وطيف ترددات الراديو للانتقال الأرضي، وطيف ترددات الراديو للانتقال عبر القمر الصناعي. تنقسم الأوساط المادية إلى قسمين رئيسين: أوساط موجّهة (Guided) وأوساط غير موجّهة (Unguided). في

الأوساط الموجّهة يتم توجيه الموجات أثناء انتقالها بواسطة وسط صلب، كالألياف الضوئية، أو زوج مجدول من أسلاك النحاس، أو كبل محوري. وفي حالة الأوساط المادية غير الموجّهة، تنتقل الموجات في الجو وفي الفضاء الخارجي، كما في الشبكات المحلية اللاسلكية أو قنوات القمر الصناعي الرقمية.

ولكن قبل أن ندخل في تفاصيل خصائص الأنواع المختلفة للأوساط المادية، دعنا نقول بضع كلمات حول تكلفتها. غالباً ما تعتبر التكلفة الفعلية للوصلة المادية (سواء كانت سلكاً نحاسياً، أو كبل ألياف ضوئية، أو أي نوع آخر) ضئيلة بالمقارنة مع بنود التكلفة الأخرى التي يتضمنها إنشاء شبكة، وعلى وجه الخصوص يمكن أن تكون تكلفة الأيدي العاملة اللازمة لتركيب الوصلة المادية أضعاف تكلفة المواد المطلوبة. لهذا السبب تقوم العديد من شركات البناء بتركيب زوج أسلاك مجدولة وليفة ضوئية وكبل محوري في كل غرفة من غرف البنايات الجديدة. حتى لو استخدم وسط واحد من هذه الأوساط في بداية الأمر، هناك احتمال كبير للاحتياج إلى وسط آخر في المستقبل القريب، وبهذا يتم توفير كلفة مد أسلاك إضافية في المستقبل.

زوج أسلاك النحاس المجدولة

يعتبر زوج أسلاك النحاس المجدولة أرخص وسائط نقل البيانات الموجّهة وأكثرها انتشاراً. ومنذ أكثر من مائة عام وشبكات الهاتف تستخدم هذا الوسط. وفي الحقيقة فإن أكثر من 99٪ من الوصلات السلكية من جهاز الهاتف إلى سنترال الهاتف المحلي تستخدم أزواج أسلاك نحاسية مجدولة، ولاشك أن معظمنا قد رأى زوج الأسلاك المجدول في منازلنا ومحيط عملنا. يتألف الزوج المجدول من سلكين نحاسيين معزولين، قطر كلٍّ منهما حوالي 1 ملم، وهما مجدولان بشكلٍ حلزوني منتظم حول بعضهما البعض لتقليل التداخل الكهرومغناطيسي الناجم عن أزواج الأسلاك المماثلة القريبة. عادةً ما يُجمع عدد من تلك الأزواج معاً على شكل كبل وذلك بتغليفها ضمن درعٍ واق، ويمثل كل زوج منها وصلة اتصال مستقلة. أما أزواج الأسلاك المجدولة المكشوفة (UTP)

فتستخدم بكثرة في شبكات الحاسب داخل البنايات (أي في الشبكات المحلية (LAN)). تتراوح معدلات نقل البيانات على تلك الشبكات الآن ما بين 10 ميجابت/ثانية إلى 1 جيجابت/ثانية، حيث تعتمد السرعة التي يمكن تحقيقها على سمك السلك والمسافة بين المرسل والمستقبل.

عندما ظهرت تقنية الألياف الضوئية في الثمانينيات، استخف الكثير من الناس بزوج الأسلاك المجدولة بسبب سرعته المنخفضة نسبياً في نقل البيانات؛ بل إن البعض تنبأ بأن تحل تقنية الألياف الضوئية محل تلك الأسلاك بالكامل. لكن زوج الأسلاك المجدولة أثبت أنه لا يستسلم بسهولة!. بوسع التقنيات الحديثة من تلك الأسلاك، كالنوعية UTP CAT 5، نقل البيانات بمعدلات تصل إلى 1 جيجابت/ثانية عبر مسافات تصل إلى مائة متر. في النهاية، أثبت زوج الأسلاك المجدول جدارته ولا يزال هو الحل المهيمن لربط شبكات البيانات المحلية السريعة.

كما ذكرنا سابقاً، يُستخدم زوج الأسلاك المجدول بكثرة أيضاً في شبكات الوصول السكني للإنترنت. رأينا أن تقنية مودم الهاتف توفر سرعات تصل إلى 56 كيلوبت/ثانية على تلك الأسلاك. رأينا أيضاً أن خط المشترِك الرقمي (DSL) مكّن المستخدمين في المنازل من الوصول للإنترنت بسرعات تتجاوز 6 ميجابت/ثانية على زوج الأسلاك المجدول (للمستخدمين القريبين من مودم موفر خدمة الإنترنت).

الكبلات المحورية

يتألف الكبل المحوري من موصلين من النحاس، مثله في ذلك مثل زوج الأسلاك المجدول، مع الفارق أن موصلي الكبل المحوري لهما نفس المركز بدلاً من كونهما متوازيين. بهذا التركيب والعزل الخاص والحجب يستطيع الكبل المحوري نقل البيانات بسرعات أعلى. يكثر استخدام هذا الكبل في أنظمة تلفزيون الكبل (Cable TV). وكما رأينا سابقاً، فقد تم مؤخراً دمج أنظمة تلفزيون الكبل بأجهزة مودم الكبل لتزويد المستخدمين في المنازل بوصول

للإنترنت بسرعات تصل إلى 1 ميجابت/ثانية أو أكثر. في تلك الأنظمة يقوم المرسل بنقل الإشارة الرقمية إلى نطاق ترددات معين (بواسطة ما يسمى بالتعديل أو التضمين (modulation)) وإرسال الإشارة التناظرية الناتجة إلى مُستقبل أو أكثر. يمكن استخدام الكبل المحوري كوسط مشترك لنقل البيانات، وعلى وجه التحديد يمكن توصيل عددٍ من الأنظمة الطرفية مباشرةً إلى الكبل، حيث يستقبل كل نظام طرفي منها كل ما يرسله أي نظام طرفي آخر.

الألياف الضوئية

الليفة الضوئية عبارة عن وسط مرن يمرر نبضات الضوء، حيث تمثل كل نبضة بت بيانات. بوسع ليفة ضوئية واحدة نقل البيانات بمعدلات فائقة تصل إلى عشرات بل مئات الجيجابت/ثانية. تمتاز الألياف الضوئية بمقاومتها للتداخل الكهرومغناطيسي، وقلة اضمحلال الإشارة المارة بها بشكلٍ كبير بحيث يمكن استخدامها لمسافات تصل إلى 100 كيلومتر من غير حاجة إلى مضخمات، كما أنه من الصعوبة بمكان التنصت على محتوى البيانات المارة بها. لقد جعلت تلك الخصائص من الألياف الضوئية وسط الإرسال الموجّه المفضّل لنقل البيانات بكميات كبيرة لمسافات طويلة، وخصوصاً للوصلات عبر البحار. والآن تستخدم العديد من وصلات المسافات الطويلة في الولايات المتّحدة الألياف الضوئية فقط، كما يسود استخدام الألياف الضوئية في العمود الفقري لشبكة الإنترنت. ومع ذلك فإن الكلفة العالية لتجهيزات الشبكات الضوئية كأجهزة الإرسال والاستقبال والمحوّلات قد أعاقت انتشارها للمسافات القصيرة كشبكات البيانات المحلية أو في المنازل كشبكات الوصول السكني. تتراوح سرعات الوصلات القياسية بتقنية الناقل الضوئي (OC) ما بين 51.8 ميجابت/ثانية و39.8 جيجابت/ثانية. تعرف هذه المواصفات غالباً باسم OC-n، حيث تساوي سرعة الوصلة حاصل ضرب n في 51.8 ميجابت/ثانية. تتضمن المعايير المستخدمة حالياً OC-1، OC-3، OC-12، OC-24، OC-48، OC-96، OC-192، OC-768. لتغطية

أكثر تفصيلاً للسمات المختلفة لشبكات الألياف الضوئية انظر [IEC Optical 2007; Goralski 2001; Ramaswami 1998; Mukherjee 1997].

قنوات الراديو الأرضية

تحمل قنوات الراديو الإشارات في طيف الترددات الكهرومغناطيسية، وتعتبر طريقة جذابة لنقل البيانات لأنها لا تحتاج لتمديد أي وسط مادي، ويمكنها اختراق الجدران، وتوفر الوصول لمستخدم نقّال، كما يمكن استخدامها لنقل الإشارات لمسافات طويلة. تعتمد خصائص قناة الراديو بشكلٍ ملحوظ على محيط انتشار الموجات والمسافة المقطوعة. تحدد الاعتبارات البيئية مدى اضمحلال الإشارة وخفوتها بالحجب (shadow fading) (واللذان ينقصان – على الترتيب – من قوة الإشارة بانتقالها لمسافة معينة، وسريانها حول أو خلال الأجسام التي تعوق انتشارها)، وكذلك ضعف الإشارة بسبب وصولها عبر مسارات متعدّدة (multi-path fading) (نتيجة انعكاس الموجات على الأجسام المختلفة)، والتداخل (interference) (بسبب الإشارات الأخرى المرسلة وغيرها من الإشارات الكهرومغناطيسية).

يمكن تصنيف قنوات الراديو الأرضية بشكلٍ عام إلى فئتين: تلك التي تُستخدم في المناطق المحلية لتغطي مسافات من عشرة أمتار إلى بضع مئات من الأمتار؛ وتلك التي تستخدم في مناطق واسعة لتغطي عشرات الكيلومترات. تستخدم تقنيات شبكات البيانات المحلية اللاسلكية التي تناولناها في الجزء 1-2-2 قنوات راديو لمناطق محلية، في حين تستخدم تقنيات الوصول الخلوية قنوات راديو لمناطق واسعة. وسوف نتناول قنوات الراديو بالتفصيل في الفصل السادس.

قنوات الراديو عن طريق الأقمار الصناعية

يقوم القمر الصناعي (satellite) بربط اثنين أو أكثر من محطات الميكروويف الأرضية للإرسال والاستقبال. يستقبل القمر الصناعي الإرسال على نطاق معين من الترددات، حيث يجدّد الإشارة المستقبلة باستخدام مكرّر (repeater)، ثم يرسل

الإشارة على نطاق تردد آخر. هناك نوعان من الأقمار الصناعية المستخدمة حالياً: أقمار ثابتة بالنسبة للكرة الأرضية (geostationary)، وأقمار تدور حول الأرض في مدار منخفض (LEO). تبقى الأقمار الصناعية من نوع geostationary بشكلٍ دائم فوق نفس البقعة على الأرض. ولتحقيق ذلك التواجد الثابت يتم وضع القمر الصناعي في مدار يبعد 36000 كيلومتر فوق سطح الأرض. تتسبب المسافة الضخمة المقطوعة من المحطة الأرضية إلى القمر الصناعي ثم العودة إلى المحطة الأرضية في تأخير كبير للإشارة يصل إلى حوالى 280 ميللي ثانية. ومع ذلك فوصلات القمر الصناعي التي يمكن أن تعمل بسرعات تصل إلى مئات الميجابت/ثانية، غالباً ما تستخدم في شبكات الهاتف وفي العمود الفقري للإنترنت. كما ذكرنا في الجزء 1-2-2، تُستخدم وصلات الأقمار الصناعية أيضاً على نحو متزايد لتحقيق وصول سكني عالي السرعة بالإنترنت في المناطق التي لا تتوفر فيها وسيلة أخرى للوصول بواسطة تقنيات DSL أو مودم الكبل.

تطلق الأقمار الصناعية من نوع LEO في مدارات أقرب بكثير لسطح الأرض، ولا تبقى بشكلٍ دائم فوق بقعة واحدة على الأرض، فهى تدور حول الأرض (تماماً كما يفعل القمر الطبيعى) ويمكنها الاتصال ببعضها البعض علاوةً على اتصالها بالمحطات الأرضية. للحصول على تغطية متصلة لمنطقة ما، ينبغي وضع العديد من تلك الأقمار الصناعية في مدارات لها حول الأرض. وجدير بالذكر أنه يجري حالياً تطوير العديد من أنظمة الاتصال على ارتفاع منخفض. تقوم صفحة الويب Lloyd لأبراج الأقمار الصناعية [Wood 2007] بتوفير وجمع المعلومات عن نظم أبراج الأقمار الصناعية للاتصالات. ويتوقع استخدام تقنية LEO للأقمار الصناعية للوصول للإنترنت في وقت ما في المستقبل.

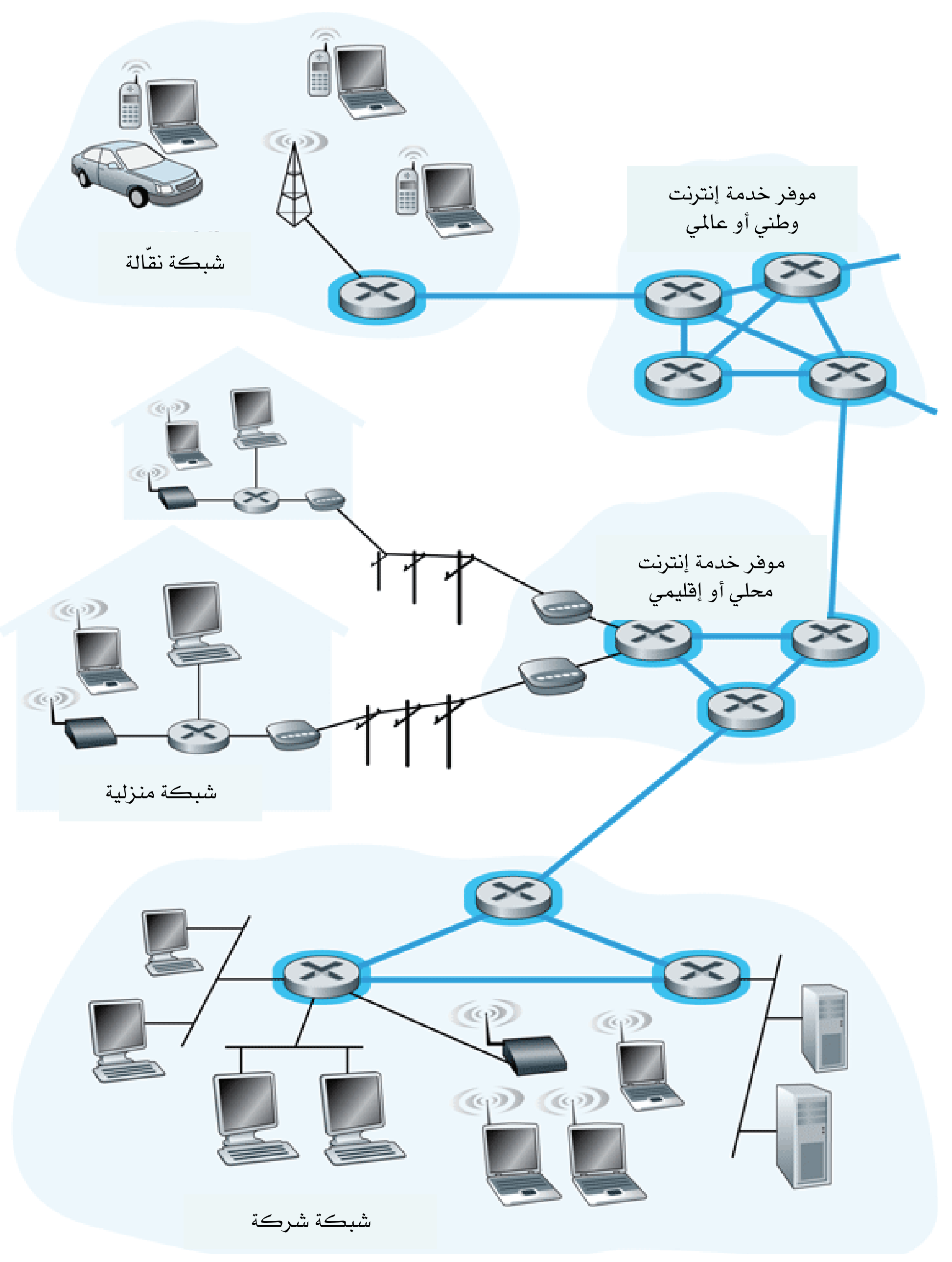
3-1 قلب الشبكة

بعد أن تناولنا حافة الإنترنت، دعنا الآن نغوص أكثر داخل قلب الشبكة؛ تلك المجموعة المتشابكة من محوّلات الرزم والوصلات التي تربط الأنظمة الطرفية الموصلة على حواف الإنترنت. يوضح الشكل 1-7 قلب الشبكة بخطوط سميكة.

1-3-1 تحويل الدوائر وتحويل الرزم

هناك طريقتان أساسيتان لنقل البيانات عبر شبكة مكونة من وصلات ومحوّلات: تحويل الدوائر (circuit switching) وتحويل رزم البيانات (packet switching). في الشبكات التي تعمل بتحويل الدوائر، تبقى كافة موارد الشبكة اللازمة على طول الطريق لتحقيق الاتصال بين الأنظمة الطرفية (كالمخازن المؤقتة ومعدلات الإرسال على الوصلات) محجوزة طوال جلسة الاتصال بين تلك الأنظمة. أما في شبكات تحويل الرزم فلا يتم حجز تلك الموارد وإنما تَستخدِم الرسائل تلك الموارد أثناء الجلسة حسب الطلب، ومن ثم فقد تحتاج الرسائل لأن تنتظر (تقف في الصف) لاستخدام وصلة اتصال. كمثال بسيط، لنأخذ بعين الاعتبار مطعمين، أحدهما يتطلب حجزاً والآخر لا يتطلب حجوزات ولا يقبلها. في حالة المطعم الأول، علينا أن نتحمل إزعاج عمل الحجز قبل أن نترك المنزل. ولكن عندما نصل إلى المطعم سيكون بوسعنا، من حيث المبدأ، أن نتحدث فوراً مع النادل ونطلب وجبة طعامنا. بالنسبة للمطعم الذي لا يتطلب حجوزات، لن نكون بحاجة لعمل حجز مسبق لمنضدة، .ولكن عندما نصل إلى المطعم، قد نحتاج للانتظار حتى تفرغ منضدة قبل أن نتمكن من الاتصال بالنادل.

تمثل شبكات الهاتف المنتشرة في كل مكان أمثلة لشبكات تحويل الدوائر. تأمّل ما يحدث عندما يرغب شخص في إرسال معلومات (صوت أو فاكس) إلى شخص آخر عبر شبكة الهاتف. قبل أن يتمكن المرسل من إرسال المعلومات، يتعين على الشبكة تجهيز وصلة مادية بين المرسل والمستقبل تعدّ



الشكل 1-7 قلب الشبكة.

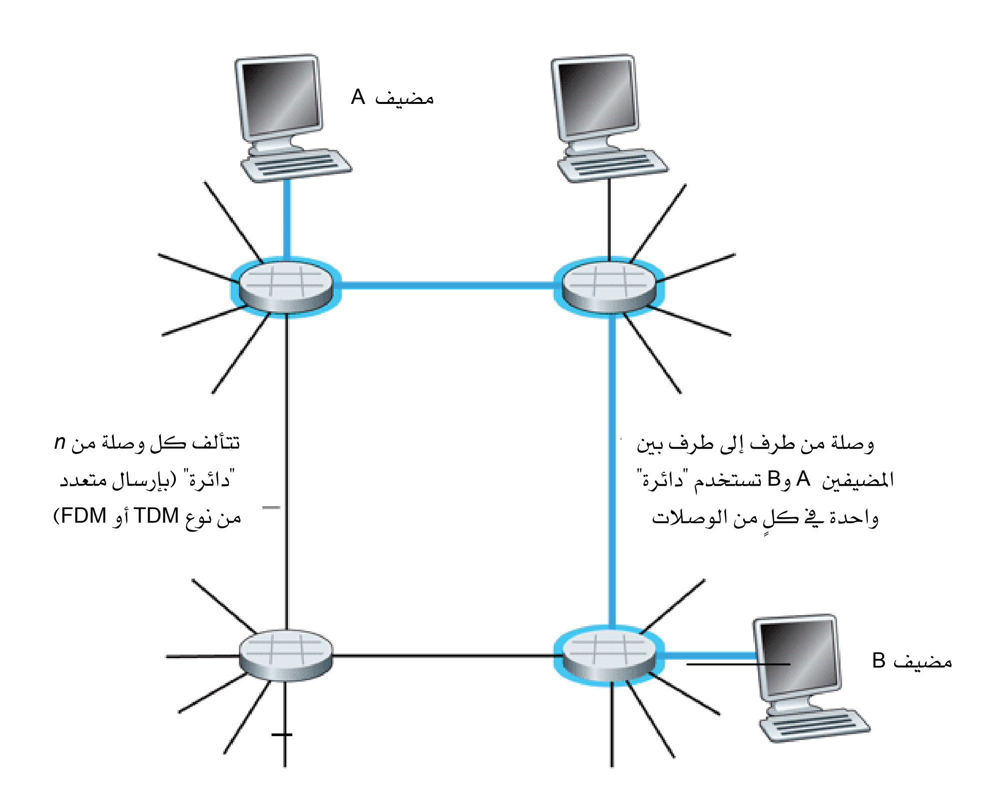
بمثابة وصلة موثوقة وتبقى كل المحوّلات الموجودة على مسار تلك الوصلة موصّلة طوال مدة جلسة الاتصال. في مفردات شبكات الهاتف، يُطلق على هذه الوصلة دائرة (circuit). وعندما تقوم الشبكة بتجهيز دائرةٍ للاتصال، تقوم أيضاً بحجز معدلٍ ثابتٍ لإرسال البيانات على وصلات الشبكة طوال المدة. نظراً لأن الحيز الترددي اللازم لتوصيل المرسل بالمستقبل قد تم حجزه مسبقاً، يكون بوسع المرسل بث البيانات إلى المستقبل بهذا المعدل الثابت.

أما إنترنت اليوم فتُعدّ مثالاً نموذجياً لشبكات تحويل الرزم. تأمل ما يحدث عندما يحتاج مضيف لإرسال رزمة إلى مضيف آخر على الإنترنت. كما هو الحال في أسلوب تحويل الدوائر، يتم إرسال الرزمة عبر سلسلة من وصلات الاتصال. ولكن بطريقة تحويل الرزم ترسل الرزمة عبر الشبكة بدون حجز أي حيز ترددى (bandwidth) على الإطلاق. فإذا صادف وكانت إحدى الوصلات مزدحمة لأن رزماً أخرى تستخدم نفس الوصلة في ذلك الوقت، ستضطر الرزمة للانتظار في المخزن المؤقت في ناحية الإرسال من الوصلة، ومن ثم تعاني تأخيراً. تبذل الإنترنت أفضل جهد لتسليم الرزم في وقت مناسب، ولكنها لا تعطي أي ضمانات (يطلق على نوع الخدمة هذه خدمة أفضل جهد (best-effort service)).

ليست كل شبكات الاتصال من النوع الذي يمكن تصنيفه بسهولة ودقة إلى شبكات بتحويل الدوائر أو شبكات بتحويل الرزم، ومع ذلك يُعدّ هذا التصنيف الأساسي نقطة بداية ممتازة لفهم تقنية شبكات الاتصال.

تحويل الدوائر

هذا كتاب عن شبكات الحاسب والإنترنت وتحويل رزم البيانات، وليس كتاباً عن شبكات الهاتف وتحويل الدوائر. ومع ذلك فمن المهم فهم لماذا تستخدم الإنترنت وشبكات الحاسب الأخرى أسلوب تحويل الرزم بدلاً من الطريقة التقليدية لتحويل الدوائر والمتبعة في شبكات الهاتف. لهذا السبب سنلقي هنا نظرة عامة مختصرة على تحويل الدوائر.



الشكل 1-8 شبكة بسيطة بتحويل الدوائر تتكون من أربعة محوّلات وأربع وصلات.

يوضح الشكل 1-8 شبكة تعمل بتحويل الدوائر. في هذه الشبكة توصّل محوّلات الدوائر الأربعة ببعضها البعض عن طريق أربع وصلات. كل وصلة من هذه الوصلات تضم n دائرة، وعليه فبوسع كل وصلة دعم n توصيلة في نفس الوقت. توصل أجهزة المضيفات (كالحواسب الشخصية ومحطات العمل) بشكلٍ مباشر بأحد المحوّلات. عندما يحتاج مضيفان للاتصال، تقوم الشبكة بتجهيز توصيلة من طرف إلى طرف تكرّسها فقط للاتصال بين هذين المضيفين. (بالطبع هناك إمكانية لعمل اتصالات بين أكثر من جهازين في نفس الوقت، ولكن لتبسيط الأمور دعنا نفترض الآن أن هناك مضيفين اثنين فقط لكل اتصال).

وهكذا، فلكي يتمكّن مضيف A من إرسال رسالة إلى مضيف B يتعين على الشبكة أولاً حجز دائرة واحدة على كل من الوصلتين بينهما. نظراً لأن كل وصلة تضم n دائرة، فإن كل توصيلة مستعملة لتحقيق الاتصال المطلوب من طرف إلى طرف ستستحوذ على جزء مقداره ($\frac{1}{n}$) من الحيز الترددي الكلي للوصلة طوال مدة الاتصال.

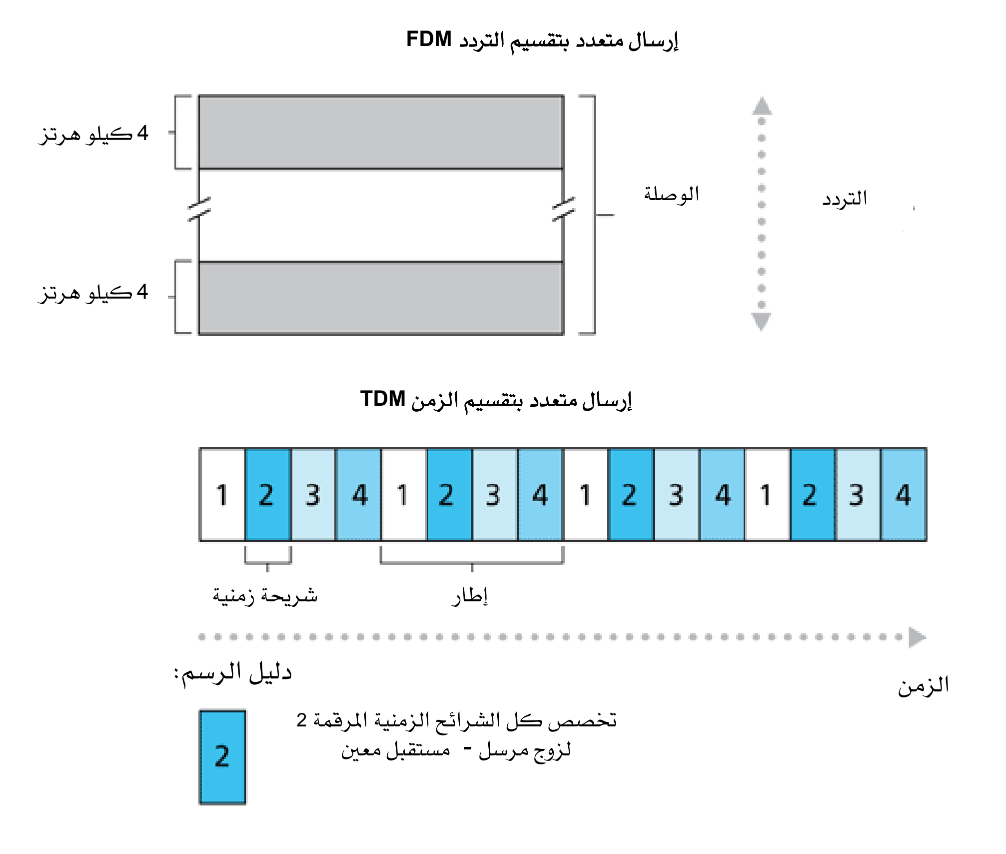
الإرسال المتعدد في شبكات تحويل الدوائر

كل دائرة من الدوائر التي توفرها الوصلة يمكن الحصول عليها إمّا بالإرسال المتعدد بتقسيم التردد (FDM) أو الإرسال المتعدد بتقسيم الوقت (TDM). في حالة FDM يتم تقسيم مجال الترددات المخصص للوصلة بين دوائر الاتصال التي توفرها الوصلة. وبشكلٍ أكثر تحديداً، تخصص الوصلة نطاق ترددات لكل اتصال ويبقى كذلك طوال مدة الاتصال. في شبكات الهاتف غالباً ما يكون نطاق التردد هذا عرضه 4 كيلوهرتز (أي 4000 ذبذبة/ثانية أو 4000 دورة/ثانية). تستخدم محطات إذاعة FM أيضاً أسلوب FDM للاشتراك في طيف التردد مابين 88 ميجاهيرتز و108 ميجاهيرتز، حيث يخصص لكل محطة نطاق تردد معين.

تقوم وصلة TDM بتقسيم الوقت إلى إطارات (frames) ثابتة المدّة، وكل إطار يقسّم إلى عدد ثابت من الفترات أو الشرائح الزمنية (time slots). عندما تُجهز الشبكة توصيلة عبر الوصلة، فإنها تقوم بتخصيص إحدى تلك الشرائح الزمنية في كل إطار لخدمة ذلك الاتصال، وتبقى تلك الشرائح الزمنية مكرّسة فقط لنقل البيانات الخاصة بذلك الاتصال إلى أن ينتهي.

يوضح الشكل 1-9 أسلوبي FDM وTDM على وصلة شبكة معينة توفر أربع دوائر. في حالة FDM يتم تجزئة حيز الترددات إلى أربعة نطاقات، عرض كل منها 4 كيلوهرتز. في حالة TDM يتم تجزئة الوقت إلى إطارات يضم كلٌّ منها أربع شرائح زمنية. تخصص لكل دائرة نفس الشريحة الزمنية في الإطارات المتتابعة ويُحسب معدل إرسال البيانات لدائرة واحدة بضرب معدل إرسال

الإطارات في عدد بتات البيانات التي تُرسل خلال الشريحة الزمنية المخصصة للدائرة. على سبيل المثال إذا كانت الوصلة ترسل 8000 إطار/ثانية وكل فترة زمنية تضم 8 بتات، يكون معدل إرسال البيانات لدائرة واحدة 64 كيلوبت/ثانية.



الشكل 1-9 في حالة الإرسال المتعدد بتقسيم التردد (FDM) يخصص لكل دائرة جزء من الحيز الترددي. في حالة الإرسال المتعدد بتقسيم الزمن (TDM) تحصل كل دائرة على حيز التردد بأكمله بشكلٍ دوري لفترة قصيرة (أي أثناء الشريحة الزمنية الخاصة بها).

يرى أنصار طريقة تحويل رزم البيانات أن تحويل الدوائر يسبب إضاعة موارد الشبكة، حيث إن الدوائر المكرّسة تبقى عاطلة أثناء فترات الصمت (عند عدم إرسال بيانات). فعلى سبيل المثال عندما يتوقف شخص عن الكلام أثناء مكالمة هاتفية، فإن موارد الشبكة (الشرائح الزمنية أو نطاق التردد المخصص له على الوصلات على طول مسار الاتصال) لن يتسنى استخدامها لخدمة التوصيلات الأخرى، حيث تبقى محجوزة له طوال مدة المكالمة.

لنأخذ مثالاً آخر يبين كيف يمكن لتلك الموارد أن تبقى غير مستغلة، لنأخذ بعين الاعتبار أخصّائي أشعة يستخدم شبكة بتحويل الدوائر للوصول عن بعد لمجموعة من صور الأشعة السينية. يقوم الأخصّائي بإعداد توصيلة، ثم يطلب صورة، ويقوم بتأملها وفحصها، وبعد ذلك يطلب صورة أخرى. تبقى موارد الشبكة مخصصة للاتصال طوال جلسة الفحص ولكنها لا تستعمل (أي تضيع) أثناء فترات تأمل الأخصّائي لصورة الأشعة. ويشير أنصار تحويل الرزم أيضاً الى أن تجهيز دوائر من طرف إلى طرف وحجز عرض نطاق من طرف إلى طرف هي عمليات صعبة وتتطلب برمجيات معقدة لتنسيق عمل المحوّلات على طول مسار التوصيلة من طرف إلى طرف.

قبل أن ننهي مناقشتنا لشبكات تحويل الدوائر، دعنا نعطي مثالاً عددياً لتسليط مزيد من الضوء على الموضوع. دعنا نحسب الوقت اللازم لإرسال ملف حجمه 640 كيلوبت من المضيف A إلى المضيف B على شبكة تعمل بتحويل الدوائر. افترض أن كل الوصلات في الشبكة تستخدم أسلوب TDM بـ 24 شريحة زمنية لكل إطار وبمعدل كلي لإرسال البيانات قدره 1.536 ميجابت/ثانية. افترض أيضاً أن تجهيز دائرة من طرف إلى طرف قبل بدء إرسال الملف يستغرق 500 ميللي ثانية. كم من الوقت سيستغرق إرسال الملف؟ كل دائرة من الدوائر الأربعة والعشرين سيخصص لها معدل إرسال قدره 1.536 ميجابت/ثانية ÷ 24 = 64 كيلوبت/ثانية، لذا يحتاج إرسال بيانات الملف إلى 640 كيلوبت ÷ (64 كيلوبت/ثانية) = 10 ثوانٍ. نضيف إلى تلك المدة الوقت اللازم لتجهيز الدائرة فنحصل على 10.5 ثانيةً كزمن كلي لإرسال الملف. لاحظ أن زمن إرسال البيانات لا يعتمد على عدد الوصلات المستخدمة، فهو 10

ثوانٍ سواء مرّت الدائرة عبر وصلة واحدة أو عبر مائة وصلة. (يتضمّن التأخير الفعلي من طرف إلى طرف أيضاً زمناً إضافياً للانتقال؛ انظر الجزء 1-4).

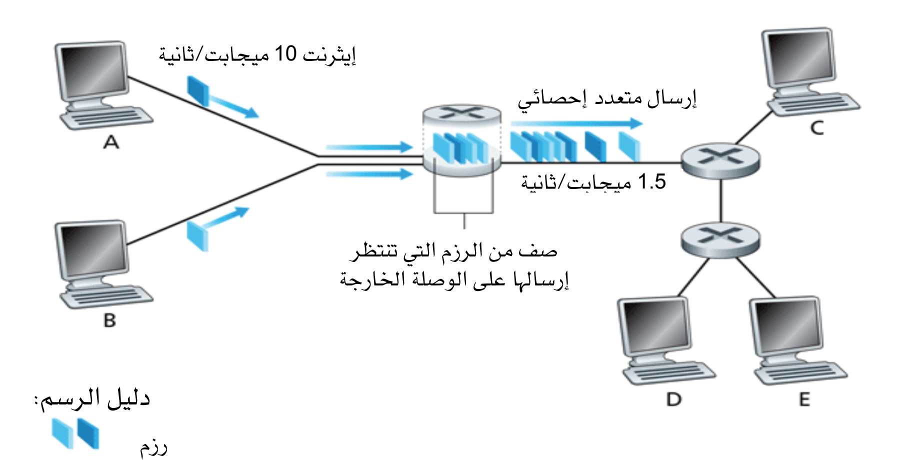
تحويل الرزم

تتبادل التطبيقات الموزّعة على الشبكة الرسائل لإنجاز مهماتها. يمكن أن تتضمن الرسائل أي شئ يقرره مصممو البروتوكول. يمكن أن تؤدي تلك الرسائل وظائف تحكم (على سبيل المثال، رسالة مرحباً في مثالنا السابق للسؤال عن الوقت). كما يمكن أن تحتوي على بيانات، كرسالة بريد إلكتروني، أو صورة JPEG، أو ملف MP3. في شبكات الحاسب الحديثة، يقوم مصدر البيانات بتجزئة الرسائل الطويلة إلى قطع أصغر من البيانات تعرف بالرزم. تنتقل كلٌّ من تلك الرزم بين المصدر والوجهة عبر وصلات الاتصال ومحوّلات الرزم (والتى يوجد منها نوعان سائدان هما الموجّهات ومحوّلات طبقة ربط البيانات). يتم إرسال بتات الرزم على كل وصلة اتصال بمعدل يساوي المعدل الكامل لإرسال البيانات على الوصلة.

تستخدم معظم محوّلات الرزم أسلوب التخزين والإرسال (store-and-forward) عند مدخل كل وصلة اتصال. يتطلب هذا الأسلوب أن يقوم المحوّل باستقبال الرزمة بالكامل قبل أن يبدأ بإرسال أول بت منها على الوصلة الخارجة. وعليه فإن أسلوب التخزين والإرسال يصاحبه تأخيرٌ عند مدخل كل وصلة على طول مسار الرزمة من المصدر إلى الوجهة النهائية. لنحسب كم يستغرق إرسال رزمة حجمها L بتات من مضيف إلى آخر عبر شبكة تستخدم تحويل الرزم. لنفترض أن هناك Q وصلة بين المضيفين، وأن معدل إرسال البيانات على كلٍّ منها هو R بت/ثانية. افترض أن تلك الرزمة هي الرزمة الوحيدة في الشبكة. يجب إرسال الرزمة أولاً على الوصلة الأولى ببثها من المضيف A وهذا يستغرق زمناً قدره $\frac{L}{R}$ ثانية. ينبغي بعد ذلك إرسالها عبرالوصلات $(Q - 1)$ المتبقية مما يتطلب تخزينها وإرسالها $(Q - 1)$ مرة، وبتأخير قدره $\frac{L}{R}$ ثانية لكل مرة. ومن ثم يكون التأخير الكلي $\frac{QL}{R}$.

يرتبط كل محوّل رزم بعدة وصلات اتصال، ولكل وصلة ملحقة به يحتفظ المحوّل بمخزن خرج مؤقت (يعرف أيضاً بصف خرج) لتخزين الرزم التي سيقوم المحوّل بإرسالها على تلك الوصلة. تلعب مخازن الخرج المؤقتة تلك دوراً رئيسياً في تحويل رزم البيانات. إذا وصلت إلى المحوّل رزمة لإرسالها عبر وصلة ولكنها وجدت تلك الوصلة مشغولة بإرسال رزمة أخرى، فعندئذ سيتعين عليها الانتظار في المخزن المؤقت. وهكذا، فإنه بالإضافة إلى التأخير الناجم عن أسلوب التخزين والإرسال، تعاني الرزم من تأخيرات الانتظار في صف المخزن المؤقت. هذه التأخيرات متغيرة وتعتمد على مستوى الازدحام في الشبكة. ونظراً لأن الأماكن في المخزن المؤقت محدودة، فقد تجد رزمة عند وصولها أن المخزن المؤقت مكتظ بالكامل برزم أخرى تنتظر الإرسال. في هذه الحالة سيحدث فقد لبعض الرزم، إما الرزمة الواصلة أو إحدى الرزم المنتظرة في الصف (حسب الأسلوب المتبع لإسقاط الرزم). وعودةً إلى المثال التقريبي للمطعم الذي أوردناه في موضعٍ سابق: يناظر تأخير الانتظار في الصف الوقت الذي تنتظره في بهو المطعم حتى تفرغ منضدة لاستخدامك. أما فقد رزمة فيناظر أن يطلب منك النادل معتذراً مغادرة المكان نظراً لأن هناك أناساً آخرين كثيرين قبلك ينتظرون مناضد.

يوضح الشكل 1-10 شبكة بسيطة تعمل بتحويل الرزم. في هذا الشكل وما يتبعه من أشكال تُمثّل الرزم بكتل ثلاثية الأبعاد حيث يُمثّل عرض الكتلة عدد البتات في الرزمة. كل الرزم في هذا الشكل لها نفس العرض، ومن ثم تحتوى على نفس عدد البتات. افترض أن المضيفين A وB يرسلان رزماً للمضيف E. فى البداية يرسل A وB رزمهما على وصلة إيثرنت بمعدل 10 ميجابت/ثانية إلى محوّل الرزم الأول والذى يقوم بدوره بتوجيه تلك الرزم إلى الوصلة الخارجة التى لها سرعة تساوى 1.5 ميجابت/ثانية. إذا تجاوز معدل وصول الرزم إلى المحوّل معدل إرسال الرزم على الوصلة الخارجة بسرعة 1.5 ميجابت/ثانية، فسيحدث ازدحام عندما تأخذ الرزم الواصلة في الاصطفاف في طابور مخزن الخرج المؤقت قبل أن تُرسل على تلك الوصلة. سنتناول تأخير الانتظار في الصف بتفصيل أكثر في الجزء 1-4.



الشكل 1-10 تحويل الرزم.

تحويل الرزم في مقابل تحويل الدوائر: الإرسال المتعدد الإحصائي

بعد أن تناولنا تحويل الدوائر وتحويل الرزم، دعنا نعقد مقارنة بين الأسلوبين. غالباً ما توجّه الانتقادات إلى أسلوب تحويل الرزم بأنه لا يناسب الخدمات الفورية (كالمكالمات الهاتفية ومؤتمرات الفيديو) نظراً للتأخيرات المتغيرة والتي يصعب التنبؤ بها من طرف إلى طرف (والتي تنجم بصورة رئيسية عن التأخيرات المتغيرة في الانتظار في الطوابير ويصعب التنبؤ بها بدقة). وفي المقابل يدافع أنصار تحويل الرزم بأن هذا الأسلوب: (1) يضمن مشاركة أفضل للحيز الترددي المتاح مقارنةً بأسلوب تحويل الدوائر، (2) يُعتبر أبسط وأكثر كفاءة وأقل كلفة من أسلوب تحويل الدوائر. للاطلاع على مناقشة مفيدة عن تحويل الرزم في مقابل تحويل الدوائر انظر [Molinero-Fernandez 2002]. وبشكلٍ عام قد يفضل الناس الذين لا يحبّون الإزعاج الذي يسببه الحجز في مطعم أسلوب تحويل الرزم على أسلوب تحويل الدوائر!

لماذا يُعدّ تحويل رزم البيانات أكثر كفاءة؟ دعنا نأخذ مثالاً بسيطاً. افترض أن المستخدمين يشتركون في وصلة سرعتها 1 ميجابت/ثانية، وأن كل مستخدم يتناوب بين فترات نشاط (يُنتج فيها بيانات بمعدل 100 كيلوبت/ثانية) وفترات

خمول لا ينتج فيها أي بيانات، ثم افترض أن المستخدم يكون نشطاً فقط بنسبة 10٪ من الوقت (ويحتسي قهوته على مهَل أثناء الـ 90٪ الأخرى). في نظام تحويل الدوائر، يجب حجز سعة إرسال مقدارها 100 كيلوبت/ثانية لكل مستخدم طوال الوقت. فعلى سبيل المثال مع نظام TDM لتحويل الدوائر، إذا تم تقسيم إطار زمني مدته ثانية واحدة إلى 10 فترات كلٍّ منها 100 ميللي ثانية، فسيخصص لكل مستخدم فترة واحدة في كل إطار.

وعليه، فإن الوصلة فى حالة استخدام أسلوب تحويل الدوائر يمكنها أن تدعم في نفس الوقت 10 مستخدمين فقط (أى ناتج قسمة 1 ميجابت/ثانية على 100 كيلوبت/ثانية). أما عند استخدام أسلوب تحويل رزم البيانات، فاحتمال أن يكون مستخدم بعينه نشطاً هو 0.1 (أي 10 بالمائة). بافتراض وجود 35 مستخدم، فاحتمال أن يكون هناك 11 مستخدماً أو أكثر نشطين في نفس الوقت سيكون تقريباً 0.0004 (توضح المسألة P7 كيفية حساب هذا الاحتمال). عندما يكون هناك 10 مستخدمين أو أقل نشطين في نفس الوقت (وذلك يحدث باحتمال قدره 0.9996)، يكون المعدل الكلي لوصول البيانات أقل من أو يساوي 1 ميجابت/ثانية (أي معدل خرج الوصلة). وهكذا فعندما يكون هناك 10 مستخدمين أو أقل نشطين، تتدفق رزم المستخدمين خلال الوصلة عملياً بدون تأخير كما هو الحال مع نظام تحويل الدوائر. أما عندما يكون هناك أكثر من 10 مستخدمين نشطين في نفس الوقت فإن معدل الوصول الكلي للرزم يتجاوز معدل خرج الوصلة ويأخذ صف الانتظار في النمو (يستمر هذا النمو حتى يتراجع معدل الوصول الكلي ليقل عن 1 ميجابت/ثانية، وعندها يبدأ طول صف الانتظار في التراجع). نظراً لأن احتمال وجود أكثر من 10 مستخدمين نشطين في نفس الوقت هو احتمال ضئيل في هذا المثال، فإن أسلوب تحويل رزم البيانات يعطي تقريباً نفس أداء أسلوب تحويل الدوائر، ولكنه يسمح بعدد أكبر من المستخدمين (35 بدلاً من 10، أي ثلاثة أضعاف ونصف).

دعنا نأخذ بعين الاعتبار مثالاَ بسيطاً آخر. افترض وجود 10 مستخدمين، وأن مستخدماً واحداً فقط يقوم فجأة بإنتاج ألف رزمة كل منها مكون من

1000 بت بينما يبقى المستخدمون الآخرون خاملون ولا ينتجون أي رزم. في نظام TDM بتحويل الدوائر يستخدم 10 فترات زمنية لكل إطار و1000 بت لكل فترة زمنية وسعة إرسال كلية مقدارها 1 ميجابت/ثانية، يستطيع المستخدم النشط استخدام الفترة الزمنية المخصصة له فقط في كل إطار لإرسال البيانات، بينما تبقى الفترات الزمنية التسعة الباقية في كل إطار خالية. سيحتاج المستخدم النشط إلى 10 ثوانٍ لإرسال المليون بت التي أنتجها. أما في حالة تحويل رزم البيانات فيمكن للمستخدم النشط إرسال رزمه بشكلٍ مستمر بمعدل الإرسال الكامل للوصلة (1 ميجابت/ثانية)، حيث لا يوجد مستخدمون آخرون ينتجون رزماً ينبغي إرسالها مع رزم المستخدم النشط بطريقة الإرسال المتعدد، وفي هذه الحالة سيتم إرسال كل البيانات التي أنتجها المستخدم النشط خلال ثانية واحدة فقط.

توضح الأمثلة السابقة حالتين يمكن أن يكون فيهما أداء تحويل الرزم أفضل من أداء تحويل الدوائر. كما تُظهر أيضاً الاختلاف الجوهري بين هذين الأسلوبين لاشتراك عدة مصادر للبيانات فيما بينها في معدل الإرسال على وصلة. يقوم أسلوب تحويل الدوائر بتحديد كيفية استخدام سعة الإرسال للوصلة بشكلٍ مسبق بغض النظر عن الطلب على تلك السعة، مما يؤدي إلى ضياع وقت الوصلة المخصص مسبقاً عند عدم استخدامه. وفي المقابل يقوم أسلوب تحويل الرزم بتخصيص استعمال الوصلة حسب الطلب، ومن ثم توزّع سعة الإرسال لوصلة مشتركة لكل رزمة على حدة، وفقط بين أولئك المستخدمين الذين لديهم رزم يلزم إرسالها على الوصلة. هذا الاشتراك في استخدام الموارد (resources) على أساس الطلب (بدلاً من التخصيص الثابت والمسبق لها) يطلق عليه أحياناً الإرسال المتعدد الإحصائي (statistical multiplexing).

رغم أن كلاً من تحويل الدوائر وتحويل الرزم منتشرٌ في شبكات الاتصالات اليوم، إلا أن الاتجاه السائد يميل بالتأكيد في صالح تحويل الرزم. حتى العديد من شبكات الهاتف اليوم تنتقل ببطء نحو تحويل رزم البيانات،

حيث تستخدم شبكات الهاتف غالباً تحويل الرزم في جزء الاتصالات عبر البحار، والذي يمثل الجزء الأعلى كلفةً للمكالمة الهاتفية.

1-3-2 كيف تسلك رزم البيانات طريقها عبر شبكات تحويل الرزم؟

ذكرنا سابقاً أن الموجّه يتسلم الرزمة التي وصلت إليه من إحدى وصلات الاتصال القادمة إليه ويرسلها عبر إحدى وصلات الاتصال الخارجة منه. ولكن كيف يحدد الموجّه الوصلة التي يجب أن يُرسل الرزمة من خلالها؟ في الواقع يتم ذلك بطرق مختلفة في الأنواع المختلفة من شبكات الحاسب. في هذا الفصل التمهيدي سنصف إحدى الطرق الشائعة، وبالتحديد الطريقة المستخدمة في الإنترنت.

تحمل كل رزمة تعبر شبكة الإنترنت عنوان الوجهة النهائية لها في ترويسة الرزمة، ولهذا العنوان تركيب هرمي كما هو الحال مع العناوين البريدية. عندما تصل رزمة إلى موجّه في الشبكة، يقوم الموجّه بفحص جزء من عنوان وجهتها النهائية ثم يرسلها إلى موجّه مجاور. وبتحديد أكثر، يضم كل موجّه جدول توجيه يربط ما بين عناوين الوجهات النهائية (أو أجزاء منها) والوصلات الخارجة المناظرة. عندما تصل رزمة إلى موجّه، يفحص الموجّه العنوان، ويبحث في جدوله لإيجاد الوصلة الخارجة المناظرة لعنوان الوجهة النهائية، ثم يقوم بعد ذلك بتوجيه الرزمة إلى تلك الوصلة.

لقد تعلمنا منذ وقت قصير أن الموجّه يستخدم عنوان وجهة الرزمة كدليل لجدول التوجيه ومن ثم تحديد الوصلة الخارجة الملائمة. لكن ذلك يطرح سؤالاً آخر: كيف يتم إعداد جداول التوجيه تلك؟ هل تُعد يدوياً في كل موجّه، أم أن الإنترنت تستخدم إجراءات أكثر آلية؟ سنتناول هذه القضية بالتفصيل في الفصل الرابع، ولكن لتبسيط الأمر الآن، سنكتفي هنا بذكر أن الإنترنت لديها عددٌ من بروتوكولات التوجيه الخاصة التي تستخدم لإعداد جداول التوجيه. على سبيل المثال، قد يحدد بروتوكول التوجيه المسار الأقصر بين كل موجّه والوجهات

النهائية المختلفة للرزم، ويستخدم تلك النتائج فى ضبط جداول التوجيه في الموجّهات.

تشبه عملية توجيه الرزم من طرف إلى طرف حالة سائق سيارة لا يستخدم الخرائط، ولكن بدلاً من ذلك يفضّل السؤال عن الطريق. على سبيل المثال، لنفرض أن جون يقود سيارته من فيلاديلفيا إلى منزل رقم 156 شارع Lakeside Drive في أورلندو بولاية فلوريدا. يقود جون السيارة أولاً إلى محطة بنزين في حيّه ويسأل كيف يصل إلى منزل رقم 156 شارع Lakeside Drive في أورلندو بولاية فلوريدا. يستخلص العامل في محطة البنزين جزء فلوريدا من العنوان ويخبر جون أن عليه أخذ الطريق السريع بين الولايات رقم I-95 باتجاه الجنوب، ويوصيه أن يسأل شخصاً آخر عند وصوله إلى ولاية فلوريدا. يأخذ جون الطريق I-95 جنوباً حتى يصل إلى مدينة Jacksonville فى فلوريدا، ثم يسأل عامل محطة بنزين آخر عن الطريق. يستخلص العامل جزء أورلندو من العنوان، ويخبر جون بأنه يجب أن يستمر على الطريق I-95 جنوباً حتى مدينة Daytona Beach، وبعد ذلك يسأل شخصاً آخر. في مدينة Daytona Beach يستخلص عامل محطة بنزين آخر جزءاً من العنوان (أورلندو) ويخبر جون أن عليه أخذ I-4 مباشرة إلى أورلندو. يأخذ جون I-4 ويغادره في مخرج أورلندو. هناك يذهب جون إلى عامل محطة بنزين آخر، وفي هذه المرة يستخلص العامل جزء Lakeside Drive من العنوان ويخبر جون عن الطريق الذي يجب عليه اتباعه للوصول إلى Lakeside Drive. عندما يصل جون إلى Lakeside Drive يسأل غلاماً على دراجة كيف يصل إلى وجهته. يستخلص الغلام رقم 156 من العنوان ويشير إلى مكان المنزل. يصل جون أخيراً إلى وجهته النهائية.

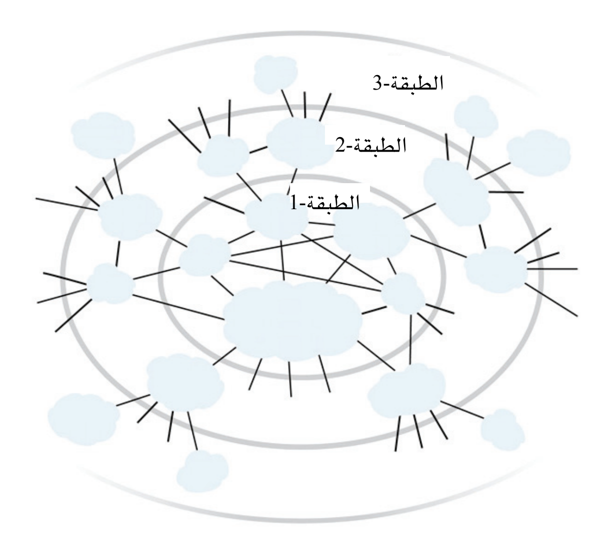
في هذا المثال التشبيهي يناظر عمال محطات البنزين والأطفال على الدراجات موجّهات الشبكة. لقد تم إعداد وتشكيل جداول التوجيه المخزنة في ذاكرتهم عبرسنوات طويلة من التجربة والخبرة.

ولكي ترى الآن بنفسك مسار رزم البيانات عبر الإنترنت من البداية إلى النهاية، يمكنك تحصيل تلك الخبرة العملية عن طريق التفاعل مع برامج تتبع

المسار (traceroute)، وذلك بالاطلاع على مناقشة عن البرنامج في الجزء 1-4 وبزيارة موقعه على الإنترنت http://www.traceroute.org.

1-3-3 موفرو خدمة الإنترنت وأعمدة الإنترنت الفقرية

رأينا فيما سبق كيف أن الأنظمة الطرفية (كحاسبات المستخدم الشخصية، والمساعدات الرقمية الشخصية، وخادمات الويب، وخادمات البريد، وما إلى ذلك) توصّل بالإنترنت عن طريق شبكات وصول. قد تكون شبكات الوصول تلك شبكات محلية سلكية أو لاسلكية (مثلاً في شركة أو مدرسة أو مكتبة)، أو مودم كبل سكني، أو وصلة DSL، أو وصلة عن طريق مودم هاتفي بموفر خدمة إنترنت سكني (مثل AOL أو MSN). غير أن توصيل المستخدمين وموفري خدمة الإنترنت إلى شبكات الوصول لا يعدو كونه جزءاً ضئيلاً من حل اللغز الكبير لتوصيل مئات الملايين من الأنظمة الطرفية ومئات الآلاف من الشبكات التي تكوّن الإنترنت. إن الإنترنت هي شبكة للربط بين شبكات، ويُعد فهم هذه الحقيقة واستيعابها بمثابة المفتاح لحل هذا اللغز.



الشكل 1-11 التوصيلات فيما بين موفري خدمة الإنترنت.

في الإنترنت العامة، ترتبط شبكات الوصول الواقعة على حافة الإنترنت ببقية الإنترنت عبر تقسيم هرمي متدرج من موفري خدمة الإنترنت، كما هو موضح في الشكل 1-11. يقع موفرو خدمة الوصول للإنترنت (على سبيل المثال شبكات الكبل وأنظمة DSL السكنية، وشبكات المودم الهاتفي مثل AOL، وشبكات الوصول اللاسلكية، وموفرو خدمة الإنترنت للجامعات والشركات عن طريق الشبكات المحلية) في أسفل هذا التدرج الهرمي. أما في أعلى القمة من هذا التدرج فيوجد عددٌ قليل نسبياً مما يسمّى بموفري خدمة الإنترنت من الطبقة -1 (tier-1) (أي الطبقة الأولى). ومع أن موفري خدمة الإنترنت من الطبقة -1 يشبهون الشبكات العادية في كثير من الأمور (فلديهم وصلات وموجّهات ويتصلون بغيرهم من الشبكات الأخرى) إلا أنهم يتميزون بأمور أخرى، فسرعات وصلاتهم غالباً ما تكون 622 ميجابت/ثانية أو أكثر، بل قد تتراوح سرعات وصلات موفري خدمة الإنترنت الكبار من هذه الطبقة ما بين 2.5 و10 جيجابت/ثانية؛ وبالتالي فإن موجّهاتهم يجب أن تكون قادرة على إرسال الرزم بسرعات عالية جداً. ويتصف موفر خدمة الإنترنت من "الطبقة -1" أيضاً بأنه:

- مرتبط مباشرة مع كل موفر آخر لخدمة الإنترنت من الطبقة -1.
- مرتبط بعدد كبير من موفري خدمة الإنترنت من الطبقة -2 وغيرهم من شبكات الزبائن.
- يغلب على تغطيته الطابع الدولي.

يُعرف موفرو خدمة الإنترنت من الطبقة -1 أيضاً بشبكات العمود الفقري للإنترنت، وتضم شركات مثل Sprint، وVerizon، وAT&T، وNTT، وLevel3، وQuest، وCable & Wireless. من المثير للانتباه أنه لا توجد مجموعة أو هيئة تمنح منزلة الطبقة -1 رسمياً، وكما يقول المثل: "إذا كنت بحاجة للسؤال عما إذا كنت تنتمي إلى مجموعة، فأغلب الظن أنك لست كذلك!"

عادةً ما يكون لموفر خدمة الإنترنت من الطبقة-2 تغطية إقليمية أو وطنية، كما أنه (وهذا مهم) يرتبط فقط بعدد قليل من موفري خدمة الإنترنت من الطبقة -1 (انظر الشكل 1-11). وعليه فلكي يصل إلى جزء كبير من الإنترنت

العالمية يتعين على موفر الخدمة من الطبقة -2 توجيه حركة بياناته عبر واحد من موفري الخدمة من الطبقة -1 المرتبط بها. يطلق على موفر خدمة الإنترنت من الطبقة -2 أنه زبون لموفر خدمة الإنترنت من الطبقة -1 المرتبط به، في حين يطلق على الأخير أنه موفر خدمة لزبائنه. توصّل العديد من الشركات والمؤسسات الكبيرة شبكتها مباشرة إلى موفر خدمة إنترنت من الطبقة -1 أو 2 ، ومن ثم تصبح زبوناً لذلك الموفر. يطالب موفر خدمة الإنترنت زبونه بأجور، تعتمد عادةً على سرعة الإرسال على الوصلة بينهما. قد يختار موفر الخدمة من الطبقة -2 أن يرتبط مباشرةً أيضاً بشبكات موفري خدمة آخرين من الطبقة-2 وفي هذه الحالة يمكن للبيانات أن تتدفق بين شبكتين من الطبقة -2 دون الحاجة للمرور عبر شبكة من الطبقة -1. يوجد أسفل موفري خدمة الإنترنت من الطبقة-2 موفرو خدمة الإنترنت من الطبقات الأدنى، والذين يوصلون إلى الإنترنت الأكبر عن طريق واحد أو أكثر من موفري الخدمة من الطبقة-2. وفي أسفل التدرج الهرمي يوجد موفرو خدمة الوصول للإنترنت. ومما يعقّد الأمور أكثر أن بعض الموفرين من الطبقة -1 هم أيضاً موفرون من الطبقة-2 (تكامل رأسي)، أي أنهم يقومون ببيع الوصول للإنترنت مباشرة إلى المستخدمين الطرفيين وموفري المحتوى، بالإضافة إلى موفري خدمة الإنترنت من الطبقات الأدنى. عندما يوصّل موفران لخدمة الإنترنت مباشرةً كل منهما بالآخر، يطلق على كلٍّ منهما أنه نظير للآخر. وللمزيد عن هذا الموضوع، راجع الدراسة [Subramanian 2002] والتى يسعى فيها الباحث للوصول إلى وصف أدق للهيكل الهرمي للإنترنت عن طريق دراسة طبوغرافية الإنترنت كدالة في علاقات توفير الخدمات للزبائن وعلاقات النظير بالنظير.

ضمن شبكة موفري خدمة الإنترنت، تُعرف النقاط التي يوصّل عندها موفر الخدمة بموفري خدمة آخرين (سواء أدنى أو أعلى أو في نفس مستوى التدرج الهرمي) بنقاط التواجد (POPs). تمثل نقطة التواجد ببساطة مجموعة من واحد أو أكثر من الموجّهات في شبكة موفر خدمة الإنترنت يتسنى توصيلها بموجّهات تابعة لموفري خدمة آخرين أو ضمن شبكات تابعة لزبائن موفر الخدمة. عادةً ما يكون لموفر الخدمة من الطبقة -1 نقاط تواجد كثيرة موزعة

عبر مناطق جغرافية مختلفة ضمن شبكته، حيث توصّل عند كل نقطة عدة شبكات للزبائن وموفري الخدمة الآخرين. لتوصيل شبكة زبون إلى نقطة تواجد خاصة بموفر خدمة، عادةً ما يقوم الزبون باستئجار وصلة عالية السرعة من موفر خدمة اتصالات كطرف ثالث، ثم يقوم بتوصيل أحد الموجّهات لديه إلى موجّه عند نقطة التواجد لدى موفر الخدمة. وعلاوة على ذلك يمكن أن يكون لاثنين من موفري الخدمة عدة نقاط تواجد يوصّلان فيها كنظيرين.

باختصار تعتبر طبوغرافية الإنترنت متشابكة ومعقّدة، حيث تشمل العشرات من موفري الخدمة من الطبقة-1 والطبقة-2 والآلاف من موفري الخدمة من الطبقات الأدنى. يتفاوت موفرو خدمة الإنترنت في المساحات التي يغطونها، فالبعض يغطي عدة قارات، والآخر يقتصر مجاله على مناطق محدودة من العالم. يرتبط موفرو الخدمة من الطبقات الأدنى بموفري خدمة من الطبقات الأعلى، بينما يرتبط موفرو الخدمة من الطبقات الأعلى ببعضهم البعض. يعتبر مستخدمو الإنترنت وموفرو المحتوى زبائن لموفري الخدمة من الطبقات الأدنى، في حين يعتبر موفرو الخدمة من الطبقات الأدنى زبائن لدى موفري الخدمة من الطبقات الأعلى.

1-4 التأخير والفقد والطاقة الإنتاجية في شبكات تحويل الرزم

ذكرنا في الجزء 1-1 أنه يمكننا اعتبار الإنترنت كبنية تحتية توفر خدمات لتطبيقات موزّعة يتم تنفيذها على أنظمة طرفية. في الحالة المثالية نود أن يكون لخدمة الإنترنت القدرة على نقل كل البيانات التي نريد نقلها من نظام طرفي إلى آخر، في الحال، وبدون أي فقد في البيانات. للأسف هذا مطلب بعيد المنال ويصعب تحقيقه على أرض الواقع. ففي الواقع نجد أن شبكات الحاسب محدودة بالضرورة من حيث طاقتها الإنتاجية (كمية البيانات التي يمكن نقلها كل ثانية) بين الأنظمة الطرفية، كما أنها تعاني من التأخير بين تلك الأنظمة، بل ومن الممكن أن تفقد بعض الرزم بالكامل. يرجع ذلك إلى قوانين الواقع المادي التي تؤدي إلى التأخير والفقد بالإضافة إلى الحد من الطاقة الإنتاجية

للشبكة. ولكن من ناحية أخرى، فلكون شبكات الحاسب تعاني من تلك المشاكل فإنه يوجد العديد من القضايا الشائقة التي تتعلق بكيفية التعامل مع تلك المشاكل ـ قضايا أكثر مما يلزم لملء منهج مقرر دراسي عن شبكات الحاسب ولتأليف المئات من أطروحات الدكتوراه في هذا المجال! سنبدأ في هذا الجزء بدراسة وتحديد التأخير والفقد والطاقة الإنتاجية في شبكات الحاسب تحديداً كمياً.

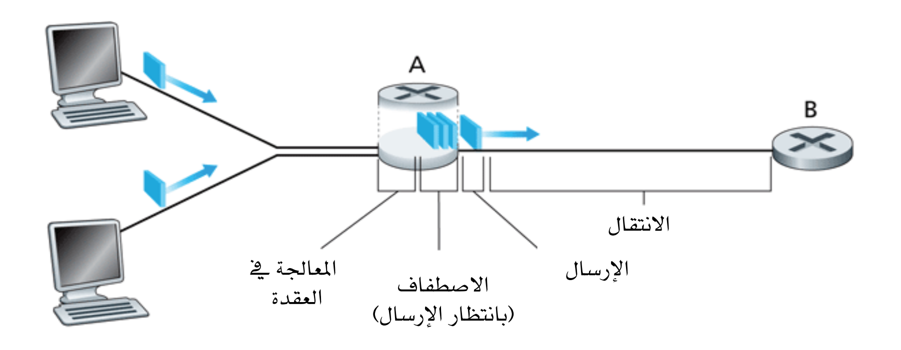
1-4-1 نظرة عامة على التأخير في شبكات تحويل الرزم

تذكّر أن رزمة البيانات تبدأ رحلتها في الشبكة من مضيف (المصدر)، وتمر عبر سلسلة من الموجّهات، لتنهي رحلتها في مضيف آخر (الوجهة النهائية). وأثناء رحلتها وهي تنتقل من عقدة على الشبكة (مضيف أو موجّه) إلى عقدة تالية (مضيف أو موجّه)، تعاني الرزمة من عدة أنواع من التأخيرات في كل عقدة على طول مسارها. من أهم تلك التأخيرات: التأخير نتيجة معالجة البيانات داخل العقدة، والتأخير بسبب انتظار الرزمة في الطوابير، والتأخير نتيجة إرسالها، والتأخير نتيجة انتقال الإشارات على الوسط المادي؛ ومجموع كل تلك التأخيرات معاً يعطي التأخير الكلي بالعقدة. ولفهم شبكات الحاسب وتحويل رزم البيانات فهماً عميقاً، علينا استيعاب طبيعة وأهمية تلك التأخيرات.

أنواع التأخيرات

دعنا نستكشف أنواع التأخيرات تلك في سياق الشكل 1-12. تُرسَل الرزمة من جهاز طرفي على الشبكة عبر الموجّه A إلى الموجّه B كجزء من مسارها بين المصدر والوجهة النهائية. هدفنا هو تحديد مكونات التأخير عند الموجّه A. لاحظ أن ذلك الموجّه لديه وصلة خارجة تؤدي إلى الموجّه B. يسبق الوصول إلى هذه الوصلة صف انتظار (يعرف أيضاً بالمخزن المؤقت). عندما تصل الرزمة إلى الموجّه A من المصدر، يقوم الموجّه بفحص ترويسة الرزمة لتحديد الوصلة الخارجة الملائمة لها ثم يوجّهها إلى تلك الوصلة. في هذا المثال، الوصلة الخارجة التي تم اختيارها للرزمة هي تلك المؤدية إلى الموجّه B. يمكن إرسال رزمة على

وصلة فقط إذا لم يكن هناك رزمة أخرى يجري إرسالها حالياً على نفس الوصلة أو إذا لم تكن هناك رزم أخرى تسبقها في صف الانتظار. إذا كانت الوصلة مشغولة حالياً أو إذا كانت هناك رزم أخرى تسبقها في الصف، فإن الرزمة الواصلة حديثاً تنضم إلى صف الانتظار.



الشكل 1-12 مكونات التأخير الكلي عند عقدة الموجّه A.

- **تأخير المعالجة**

يعتبر الوقت اللازم لفحص ترويسة الرزمة وتحديد الوصلة الخارجة التي ينبغي توجيهها إليها جزءًا من تأخير المعالجة. يمكن أن يتضمّن تأخير المعالجة أيضاً عدة عوامل أخرى كالوقت اللازم للتدقيق بحثاً عن وجود أخطاء في بتات الرزمة والتي يمكن أن تكون قد حدثت أثناء إرسالها من المصدر إلى الموجّه A. وعادةً ما يكون تأخير المعالجة في الموجّهات عالية السرعة في حدود الميكروثانية أو أقل. بعد هذه المعالجة في العقدة يقوم الموجّه بتوجيه الرزمة إلى صف الانتظار الذي يسبق الوصلة المؤدية إلى الموجّه B. سنتناول في الفصل الرابع بالتفصيل كيفية عمل الموجّه.

- **تأخير الانتظار**

تعاني الرزمة من التأخير نتيجة الانتظار في الصف حتى يحين دورها لكي تُرسَل عبر الوصلة الخارجة. ويعتمد تأخير الانتظار في الصف لرزمة بعينها على عدد الرزم التي سبق وصولها والتي اصطفت منتظرةً الإرسال عبر الوصلة نفسها. فإذا كان صف الانتظار فارغاً ولا توجد رزمة أخرى يجري إرسالها حالياً، فسيكون تأخير الانتظار بالنسبة لتلك الرزمة الواصلة صفراً. ولكن إذا كانت حركة مرور البيانات مزدحمة، ويوجد العديد من الرزم الأخرى قبلها تنتظر دورها في الإرسال، فسيكون تأخير الانتظار طويلاً. سنرى بعد قليل أن عدد الرزم المنتظرة التي يُتوقع أن تجدها رزمة واصلة هو دالة في كثافة وطبيعة حركة مرور الرزم التي تصل إلى صف الانتظار. عملياً يتراوح تأخير الانتظار من ميكروثانية إلى ميللي ثانية.

- **تأخير الإرسال**

على افتراض أن الرزم تُرسَل حسب ترتيب وصولها (أي بأسلوب الخدمة أولاً للواصل أولاً (first-come-first-served (FCFS))، كما هو المتبع عادةً في شبكات تحويل الرزم، فإن رزمتنا سترسل فقط بعد أن يتم إرسال كل الرزم التي وصلت قبلها. لنرمز لطول الرزمة بالبتات بالرمز L، ولمعدل إرسال البيانات على الوصلة من الموجّه A إلى الموجّه B بالرمز R بت/ثانية. على سبيل المثال، لوصلة إيثرنت سرعتها 10 ميجابت/ثانية فإن R = 10 ميجابت/ثانية؛ ولوصلة إيثرنت سرعتها 100 ميجابت/ثانية تكون R = 100 ميجابت/ثانية. تأخير الإرسال (والذي يعرف أيضاً بتأخير التخزين والإرسال كما بينّا في الجزء 1-3) يساوي $\frac{L}{R}$. تلك هي كمية الوقت اللازمة لدفع (أي إرسال) كل بتات الرزمة إلى الوصلة. عملياً يكون تأخير الإرسال في حدود من ميكروثانية إلى ميللي ثانية.

- **تأخير الانتقال**

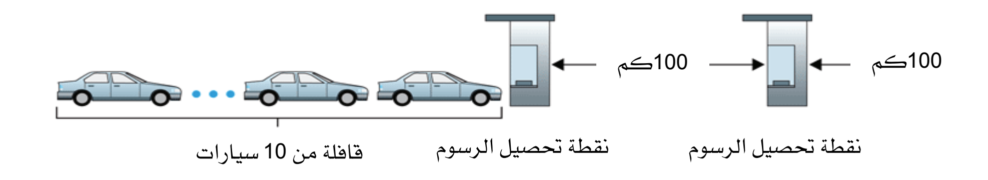
بمجرد دفع بت بيانات إلى الوصلة، فإنه يحتاج للانتقال إلى الموجّه B. ويُطلق على الوقت اللازم لانتقال البت من بداية الوصلة (عند الموجّه A) إلى نهايتها (عند الموجّه B) تأخير الانتقال. ينتقل البت بسرعة انتقال موجة الإشارة على الوصلة، وتعتمد تلك السرعة على نوع الوسط المادي للوصلة (أي ألياف ضوئية، أو زوج أسلاك نحاس مجدولة، وهكذا)، وهي في الحدود من 2×10^8 إلى 3×10^8 متر/ثانية، والتي تساوي أو تقل قليلاً عن سرعة الضوء. يُحسب تأخير الانتقال بقسمة المسافة بين موجّهين على سرعة الانتقال، أي أن تأخير الانتقال = $\frac{d}{s}$، حيث d هي المسافة بين الموجّهين، s هي سرعة الانتقال على الوصلة. بمجرد وصول البت الأخير من الرزمة إلى العقدة B على الشبكة يتم تخزينها مع كل بتات الرزمة التي وصلت قبلها على الموجّه B. يتم تكرار العملية بعد ذلك من جديد حيث يقوم الموجّه B بإرسال الرزمة إلى الموجّه التالى. ويكون تأخير الإرسال في حدود الميللي ثانية في الشبكات التي تغطي مناطق شاسعة (WAN).

مقارنة بين تأخيرات الإرسال والانتقال

أحياناً ما يجد المبتدئون فى مجال شبكات الحاسب صعوبة في فهم الاختلاف بين تأخير الإرسال وتأخير الانتقال. إن الاختلاف دقيق ولكنه مهم. فتأخير الإرسال هو كمية الوقت اللازمة لموجّه لإخراج الرزمة؛ ومن ثم فإنه دالة في طول الرزمة ومعدل إرسال البيانات على الوصلة، لكن ليس له أي علاقة بالمسافة بين الموجّهين. أما تأخير الانتقال فهو الوقت الذي يستغرقه بت واحد للانتقال من موجّه إلى الموجّه التالي عبر وصلة؛ ومن ثم فإنه دالة في المسافة بين الموجّهين وسرعة انتقال الاشارة فى الوسط المادى، ولكن ليس له أي علاقة بطول الرزمة أو معدل إرسال البيانات على الوصلة.

قد يوضّح المثال التالي مفهوم كلٍّ من تأخير الإرسال وتأخير الانتقال. لنأخذ في الاعتبار طريقاً سريعاً عليه نقاط لتحصيل الرسوم كل 100

كيلومتر، كما هو موضح في الشكل 1-13. يمكنك اعتبار أجزاء الطريق السريع ما بين نقاط التحصيل كالوصلات، ونقاط التحصيل نفسها كالموجّهات. افترض أن السيارات تسافر على الطريق السريع بسرعة 100 كيلومتر/ساعة (أي أنه عندما تترك سيارة نقطة التحصيل، فإنها تحقق تسارعاً في الحال لتصل سرعتها إلى 100 كيلومتر/ساعة وتحافظ على تلك السرعة ثابتة بين النقاط. افترض بعد ذلك أن قافلة مكونة من 10 سيارات تسافر معاً ويتبع بعضها بعضاً بنفس الترتيب. يمكنك اعتبار كل سيارة تمثل بت بيانات بينما تمثل القافلة رزمة بيانات. افترض أيضاً أن كل نقطة تحصيل تقوم بخدمة (أي إرسال) السيارات بمعدل سيارة كل 12 ثانية (أي 5 سيارات/دقيقة)، وأن السيارات كانت تسير في ساعة متأخرة من الليل بحيث لا توجد سيارات أخرى على الطريق السريع. وأخيراً لنفترض أنه حينما تصل السيارة الأولى من القافلة إلى نقطة التحصيل، فإنها تنتظر في المدخل حتى تصل السيارات التسع الأخرى وتتراص خلفها. (وبذلك يتم "تخزين" كامل القافلة عند نقطة التحصيل قبل البدء في إرسال القافلة عبر القطعة التالية من الطريق).



الشكل 1-13 مثال قافلة السيارات.

الوقت الذي تحتاجه النقطة لدفع كامل القافلة على الطريق السريع = (10 سيارات) / (5 سيارات/دقيقة) = 2 دقيقة، وهذا الوقت يناظر تأخير الإرسال للموجّه. الوقت الذي تحتاجه سيارة للسفر من لحظة خروجها من نقطة تحصيل إلى وصولها إلى النقطة التي تليها = 100 كيلومتر ÷ (100 كيلومتر/ساعة) = 1 ساعة، وهذا الوقت يناظر تأخير الانتقال. وعليه فإن الوقت الكلي من لحظة اصطفاف القافلة مخزّنةً أمام نقطة تحصيل حتى

لحظة اصطفافها مخزّنة أمام النقطة التالية هو مجموع تأخير الإرسال وتأخير الانتقال(أي 62 دقيقة في هذا المثال).

لنستطرد بعض الشيء في تأمل هذا المثال. ماذا يحدث لو أن وقت الخدمة في نقطة التحصيل كان أكبر من الوقت الذي تستغرقه السيارة في السفر بين نقطتي تحصيل؟ على سبيل المثال، افترض الآن أن السيارات تسير بسرعة 1000 كيلومتر/ساعة وأن النقطة تخدم السيارات بمعدل سيارة واحدة في الدقيقة. عندئذ يكون تأخير الانتقال بين نقطتين هو 6 دقائق، والوقت اللازم لخدمة القافلة في النقطة هو 10 دقائق. في هذه الحالة ستصل السيارات الأولى من القافلة إلى النقطة التالية قبل أن تغادر السيارات الأخيرة من القافلة النقطة الأولى. يمكن أن يحدث ذلك أيضاً في شبكات تحويل الرزم، حيث قد تصل البتات الأولى من رزمة إلى موجّه بينما العديد من البتات الباقية من الرزمة ما تزال تنتظر إرسالها من الموجّه السابق.

وكما يقال أنه إذا كانت الصورة الثابت تحكي أكثر من ألف كلمة، فإنه حريٌ بالصورة المتحركة أن تحكي أكثر من مليون كلمة! لذا يتضمن موقع الويب المصاحب لهذا الكتاب برنامج جافا تفاعلي (Java Applet) يصور بشكلٍ لطيف تأخير الإرسال وتأخير الانتقال ويقارن بينهما، وننصح القارئ بزيارة الموقع واستعراض ذلك البرنامج. لنفترض أن d_{proc}، وd_{queue}، وd_{trans}، وd_{prop} تمثل تأخير المعالجة، والانتظار، والإرسال، والانتقال على الترتيب. عندئذٍ تحدد المعادلة التالية التأخير الكلي عند كل عقدة d_{nodal} في الشبكة:

$$d_{nodal} = d_{proc} + d_{queue} + d_{trans} + d_{prop}$$

يمكن أن تتفاوت مساهمة كلٍّ من مكونات التأخير في المعادلة أعلاه بشكلٍ ملحوظ، فمثلاً قد تكون d_{prop} لوصلة بين موجّهين في نفس الحي الجامعي ضئيلة جداً بحيث تهمل (مثلاً حوالي 2 ميكروثانية).ومع ذلك يمكن أن تبلغ d_{prop} مئات الميللي ثانية بين موجّهين أرضيين موصلين عبر قمر صناعي، ومن ثم تكون بمثابة مركّبة التأخير المهيمنة في تأخير العقدة الكلي d_{nodal}.

بنفس الطريقة يمكن أن يتراوح تأثير d_{trans} من ضئيل إلى مؤثر؛ فمثلاً يكون تأثير تلك المركّبة ضئيلاً عادةً عند معدلات الإرسال ابتداءً من 10 ميجابت/ثانية فما فوق (على سبيل المثال، فى بعض أنواع الشبكات المحلية)؛ ومع ذلك يمكن أن تصل إلى مئات الميللي ثانية لرزم إنترنت كبيرة مرسلة على وصلات مودم هاتفية بسرعات بطيئة. يمكن إهمال تأخير المعالجة d_{proc} في أغلب الأحيان؛ ومع ذلك فله تأثير قوي على الطاقة الإنتاجية القصوى للموجّه (أى المعدل الأقصى لإرسال الرزم بواسطة الموجّه).

1-4-2 تأخير الانتظار وفقد الرزم

يعتبر تأخير الانتظار في الصف d_{queue} أكثر مكونات تأخير العقدة أهمية وتعقيداً في مجال شبكات الحاسب، ولهذا فقد كُتِبَت عنه آلاف الأبحاث وأُلِفَت العديد من الكتب [Bertsekas 1991; Daigle 1991; Kleinrock 1975, 1976; Ross 1995]. سنعطي هنا فقط مجرد مناقشة بديهية تجريدية للتأخير في صف الانتظار؛ وبوسع القارئ المهتم مراجعة بعض الكتب الأخرى (أو حتى كتابة أطروحة دكتوراه في الموضوع!). بخلاف التأخيرات الثلاثة الأخرى (أي d_{proc}، d_{trans}، وd_{prop})، يمكن أن يتفاوت تأخير الانتظار في الصف من رزمة إلى رزمة. فعلى سبيل المثال إذا وصلت 10 رزم إلى صف انتظار فارغ في نفس الوقت، فإن الرزمة الأولى لن تعاني في إرسالها أي انتظار في الصف، بينما ستعاني الرزمة الأخيرة المرسلة تأخير انتظار كبير نسبياً (لانتظار إرسال الرزم التسعة الأخرى). ولذا تُستخدم عند دراسة تأخير الانتظار في الصف عادةً مقاييسٌ إحصائية، كمتوسط التأخير والتباين في التأخير واحتمال تجاوز التأخير قيمة معينة.

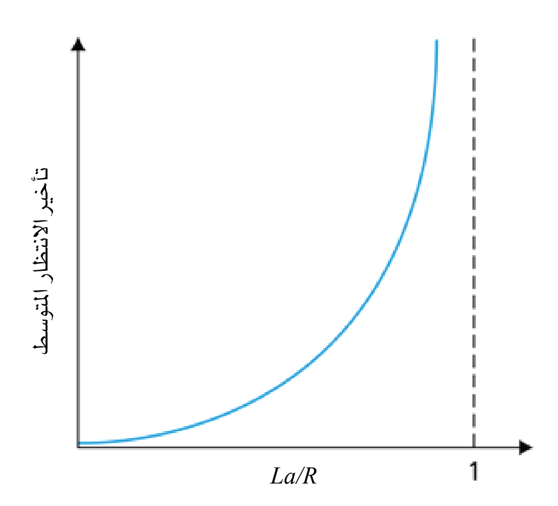
متى يكون تأخير الانتظار في الصف كبيراً ومتى يمكن إهماله؟ يعتمد الجواب عن هذا السؤال على معدل وصول الرزم إلى صف الانتظار ومعدل إرسال البيانات على الوصلة وطبيعة حركة مرور البيانات الواصلة (أي: ما إذا كانت البيانات تصل بشكلٍ دوري أو على هيئة نبضات قصيرة). لإلقاء بعض الضوء على هذا الموضوع، لنفترض أن a تمثل متوسط معدل وصول الرزم إلى صف الانتظار (ووحداتها رزمة/ثانية). تذكّر أن R هي معدل الإرسال

(بت/ثانية)؛ أي المعدل الذي يتم به دفع البتات على الوصلة، ومن ثم خروجها من صف الانتظار. لنفترض أيضاً للتبسيط أن كل الرزم الواصلة تتألف من L بت. عندئذ يكون المعدل المتوسط لوصول البتات لصف الانتظار هو Laبت/ثانية. أخيراً افترض أن المخزن المؤقت (صف الانتظار) كبير جداً، بحيث يمكن أن يستوعب عدداً لا نهائياً من البتات. تُعرف النسبة $\frac{La}{R}$ بكثافة حركة المرور، وتلعب غالباً دوراً هاماً في تقدير مدى التأخير بسبب الانتظار في الصف. فإذا كانت $\frac{La}{R}>1$، فإن المعدل المتوسط لوصول البتات لصف الانتظار يتجاوز معدل إرسال البتات من صف الانتظار، وفي هذه الحالة المؤسفة سيتجه طول صف الانتظار إلى الزيادة بدون حدّ، ويقترب تأخير الانتظار في الصف من اللانهاية!. ومن ثم فإن إحدى القواعد الذهبية في هندسة المرور هي: "صمّم نظامك بحيث لا تتجاوز كثافة المرور القيمة 1".

لنأخذ في الاعتبار الآن الحالة $\frac{La}{R}\leq 1$. هنا تؤثر طبيعة حركة مرور رزم البيانات الواصلة على تأخير الانتظار في الصف. فمثلاً إذا كانت الرزم تصل بشكلٍ دوري بمعدل رزمة واحدة كل $\frac{L}{R}$ ثانية (أي $a=\frac{R}{L}$)، فحينئذ ستصل كل رزمة إلى صف انتظار فارغ ولن يكون هناك تأخير انتظار. أما إذا وصلت الرزم على شكل نبضات ولكن بشكلٍ دوري، فيمكن أن يؤدي ذلك إلى تأخير انتظار ملحوظ في المتوسط. لنفترض على سبيل المثال أن الرزم تصل في نفس الوقت بمعدل N رزمة كل $\frac{NL}{R}$ ثانية، ففي هذه الحالة لا تعاني الرزمة الأولى من أي تأخير انتظار؛ أما الرزمة الثانية فتنتظر $\frac{L}{R}$ ثانية. وبشكلٍ عام فإن الرزمة رقم n تعاني من تأخير انتظار قدره $\frac{(n-1)L}{R}$ ثانية. وكتمرين سنترك للقارئ حساب تأخير الانتظار المتوسط في هذا المثال.

يغلب على المثالين السابقين الطابع الأكاديمي نوعاً ما، حيث افترضنا وصول الرزم بشكلٍ دوري منتظم. ففي الواقع العملي عادةً ما يكون وصول الرزم إلى صف الانتظار بشكلٍ عشوائي؛ بحيث إن الرزم الواصلة لا تتبع أي

نمط ويفصل بينها فترات عشوائية من الزمن. في هذه الحالة الأكثر واقعية، لن تكون الكمية $\frac{La}{R}$ كافية عادةً لتحديد الخواص الإحصائية لتأخير الانتظار بالكامل. ومع ذلك فهي مفيدة لتحصيل فهم بديهي لمدى التأخير. وبالتحديد، إذا كانت كثافة حركة مرور الرزم قريبة من الصفر، وكانت الرزم تصل قليلة ومتباعدة فيما بينها، فمن غير المحتمل أن تجد رزمة واصلة رزمة أخرى في صف الانتظار. وعليه فإن متوسط تأخيرالانتظار في الصف سيكون صفراً تقريباً. من الناحية الأخرى عندما تكون كثافة حركة المرور قريبة من 1، سيكون هناك فترات من الوقت يتجاوز فيها معدل الوصول قدرةَ الإرسال (بسبب الاختلافات في معدل وصول الرزم)، ويبدأ صف انتظار في التكون أثناء تلك الفترات. أما عندما يكون معدل وصول الرزم أقل من قدرة الإرسال، فسينكمش طول صف الانتظار. ومع ذلك فعندما تقترب كثافة حركة المرور من 1، يصبح صف الانتظار المتوسط أطول وأطول. يبين الشكل 1-14 طبيعة العلاقة المعتادة في الواقع العملي بين تأخير الانتظار المتوسط وكثافة حركة مرورالبيانات.



الشكل 1-14 طبيعة اعتماد تأخير الانتظار المتوسط على كثافة حركة مرور البيانات.

من السمات المهمة في الشكل 1-14 أنه عندما تقترب كثافة حركة المرور من 1، يزداد متوسط تأخير الانتظار في الصف بسرعة فائقة. أي أن زيادة الكثافة بنسبة مئوية صغيرة تؤدي إلى زيادة في التأخير بنسبة مئوية أكبر بكثير. ولعلك لاحظت هذه الظاهرة على الطريق السريع. إذا كنت تقود سيارتك بانتظام على طريق مزدحم عادةً، فإن ازدحام الطريق في العادة يعني أن كثافة حركة المرور تكون قريبة من 1. إذا طرأ شيء تسبب في زيادة حتى ولو طفيفة في حركة المرور عن المعدلات المعتادة، فإن التأخيرات التي تواجهها حركة المرور نتيجة لذلك يمكن أن تكون كبيرة.

لكي يتكون لديك تصور جيد عن ماهية تأخيرات الانتظار والعوامل المؤثرة فيها، ندعوك مرةً أخرى لزيارة موقع الويب المصاحب لهذا الكتاب، والذي يتضمن برنامج جافا تفاعلي يحاكي صف انتظار. إذا وضعت معدل وصول الرزم عالياً بما فيه الكفاية بحيث تتجاوز كثافة حركة مرور البيانات القيمة 1، فسترى كيف يأخذ صف الانتظار في النمو ببطء مع مرور الوقت.

فقد الرزم

في المناقشة السابقة، افترضنا أن صف الانتظار يتسع لعدد لانهائي من الرزم، غير أنه في واقع الأمر تكون سعة صف الانتظار الموجود على بوابة كل وصلة محدودة، علماً بأنها تعتمد كثيراً على تصميم وكلفة محوّل الرزم. ونظراً لأن سعة الانتظار في الصف محدودة، فإن تأخيرات الرزم لا تقارب ما لا نهاية في الواقع عندما تقترب كثافة حركة المرور من 1. غير أنه يمكن أن تصل رزمة لتجد صف الانتظار مملوءاً بالكامل، وعندها سيضطر الموجّه إلى إسقاط الرزمة من حسابه، أي أن تلك الرزمة ستُفقد.

يمكن أيضاً مشاهدة ظاهرة فيضان صف الانتظار هذه في المحاكاة التي يقوم بها برنامج جافا التفاعلي لصف انتظار عند كثافة لحركة المرور

أكبر من 1. من وجهة نظر نظام طرفي مرسِل، ستبدو رزمة مفقودة كرزمة أرسلت إلى قلب الشبكة ولكنها لم تخرج منها من الناحية الأخرى باتجاه وجهتها المقصودة. تزداد نسبة الرزم المفقودة بزيادة كثافة حركة المرور، ولذا فإن أداء عقدة ما على الشبكة غالباً ما يقاس، ليس فقط بدلالة التأخير الناجم، ولكن أيضاً باحتمالية فقد الرزم. كما سيتضح في تناولنا للفصول التالية، قد نضطر في حالة رزمة فُقدت لإعادة إرسالها بطريقة موثوقة أكثر (من طرف إلى طرف) لكي نضمن أن كل البيانات المطلوبة يتم نقلها جميعاً في نهاية الأمر من المصدر إلى الوجهة.

1-4-3 التأخير من طرف إلى طرف

ركّزت معالجتنا حتى الآن على التأخير عند كل عقدة على الشبكة، أي التأخير عند موجّه واحد. لنأخذ في الاعتبار الآن التأخير الكلي من المصدر إلى الوجهة. ولاستيعاب هذا المفهوم، لنفترض أن هناك N-1 موجّه بين مضيف المصدر ومضيف الوجهة. لنفترض أيضاً في البداية أن الشبكة غير مزدحمة (ومن ثم يمكن إهمال تأخيرات الانتظار)، وأن تأخير المعالجة في كلّ موجه وفي مضيف المصدر هو d_{proc}، ومعدل إرسال البيانات من كل موجّه ومن مضيف المصدر هو R بت/ثانية، وتأخير الانتقال على كل وصلة هو d_{prop}. تتراكم تأخيرات العقد لتعطي تأخيراً من طرف إلى طرف $d_{\text{end-end}}$ قدره:

$$d_{\text{end-end}} = N \times (d_{\text{proc}} + d_{\text{trans}} + d_{\text{prop}})$$

حيث $d_{trans} = \frac{L}{R}$، وL هي حجم الرزمة بالبتات.

سنترك لك كتمرين تعميم هذه المعادلة في حالة تفاوت التأخيرات عند العقد المختلفة في الشبكة وفي وجود تأخير انتظار متوسط عند كل عقدة.

برنامج تتبع المسار (traceroute)

للحصول على تدريب عملي على التأخير من طرف إلى طرف في شبكة حاسب، يمكنك استخدام برنامج تتبع المسار traceroute، وهو برنامج بسيط

يمكن تشغيله على أي مضيف بالإنترنت. عندما يحدد المستخدم اسم مضيف الوجهة، فإن هذا البرنامج في مضيف المصدر يرسل عدة رزم خاصة نحو تلك الوجهة. وبينما تشق تلك الرزم طريقها إلى الوجهة فإنها تعبر سلسلة من الموجّهات. عندما يتسلم كل موجّه إحدى تلك الرزم الخاصة، فإنه يرسل إلى المصدر رسالة قصيرة تضم اسم وعنوان ذلك الموجّه.

بشكلٍ أكثر تحديداً، لنفترض أن هناك *N*-1 موجّه بين المصدر والوجهة النهائية. في هذه الحالة سيرسل المصدر *N* رزمة خاصة إلى الشبكة، كلاً منها معنونة إلى الوجهة النهائية. ترقّم تلك الرزم الخاصة بالأرقام التسلسلية من 1 إلى *N*، حيث يمثل 1 الرزمة الأولى و*N* الرزمة الأخيرة. عندما يستلم الموجّه رقم *n* الرزمة *n*، لا يقوم بإرسال الرزمة نحو وجهتها، ولكنه بدلاً من ذلك يرسل رسالة إلى المصدر. عندما يستلم مضيف الوجهة النهائية الرزمة N، يرسل أيضاً رسالة إلى المصدر. يقوم المصدر بتسجيل الوقت الذي ينقضي بين إرسال كل رزمة وتسلم الرسالة الراجعة المناظرة لها، كما يسجّل أيضاً اسم وعنوان الموجّه (أو مضيف الوجهة النهائية) الذي أعاد الرسالة. بهذه الطريقة يمكن للمصدر إعادة تركيب المسار الذي سلكته الرزم في تدفقها من المصدر إلى الوجهة، ويكون بوسع المصدر أيضاً قياس زمن التأخير في رحلة الذهاب والإياب إلى كلٍّ من الموجّهات الموجودة بين المصدر والوجهة. في واقع الأمر يكرر برنامج تتبع المسار هذه التجربة ثلاث مرات، أي أن المصدر يرسل في الحقيقة 3N رزمة إلى الوجهة. ولتفاصيل طريقة تتبع المسار راجع المستند RFC 1393.

وإليك مثالاً للبيانات الناتجة من تنفيذ برنامج تتبع المسار، حيث يمتد المسار الذي تم تتبّعه من مضيف المصدر gaia.cs.umass.edu (في جامعة ماسوشوستس) إلى مضيف cis.poly.edu (في جامعة بوليتكنيك بروكلن) كوجهة نهائية. يتألف ناتج البرنامج من ستة أعمدة: يحتوي العمود الأول قيمة *n* والتى تمثل رقم الموجّه على طول المسار كما ذكرنا من قبل، ويعطي العمود الثاني اسم الموجّه، ويبين العمود الثالث عنوان الموجّه (بالشكل xxx.xxx.xxx.xxx)، أما الأعمدة الثلاثة الأخيرة فتتضمن ثلاثة قياسات لقيم

التأخير لرحلة الذهاب والإياب فى أوقات مختلفة. لاحظ أنه فى حالة وجود فقد للرزم في الشبكة قد يستلم المصدر أقل من ثلاث رسائل من أي من الموجّهات على المسار، عندئذ سيضع برنامج تتبع المسار نجمة مباشرة بعد رقم الموجّه المناظر ويعطي أقل من ثلاث نتائج لقيم التأخير لذلك المسار.

```
1 cs-gw (128.119.240.254) 1.009 ms 0.899 ms 0.993 ms
2 128.119.3.154 (128.119.3.154) 0.931 ms 0.441 ms 0.651 ms
3 border4-rt-gi-1-3.gw.umass.edu (128.119.2.194) 1.032 ms 0.484 ms 0.451 ms
4 acrl-ge-2-1-0.Boston.cw.net (208.172.51.129) 10.006 ms 8.150 ms 8.460 ms
5 agr4-loopback.NewYork.cw.net (206.24.194.104) 12.272 ms 14.344 ms 13.267 ms
6 acr2-loopback.NewYork.cw.net (206.24.194.62) 13.225 ms 12.292 ms 12.148 ms
7 pos10-2. core2.NewYorkl.Leve13.net (209.244.160.133) 12.218 ms 11.823 ms 11.793 ms
8 gige9-1-52.hsipaccessl.NewYorkl.Leve13.net (64.159.17.39) 13.081 ms 11.556 ms 13.297 ms
9 p0-0.polyu.bbnplanet.net (4.25.109.122) 12.716 ms 13.052 ms 12.786 ms
10 cis.poly.edu (128.238.32.126) 14.080 ms 13.035 ms 12.802 ms
```

يحتوي المسار أعلاه على تسعة موجّهات بين المصدر والوجهة النهائية. أغلب هذه الموجّهات لها اسم، ولكل منها عنوان. فعلى سبيل المثال، اسم الموجّه 3 هو border4-rt-gi-1-3.gw.umass.edu وعنوانه 128.119.2.194. بالرجوع إلى البيانات المسجلة لهذا الموجّه نفسه، نجد أنه في أوّل المحاولات الثلاث كان تأخير رحلة الذهاب والإياب بين المصدر والموجّه 1.032 ميللي ثانية، بينما كان التأخير في المحاولتين اللاحقتين 0.484 و0.451 ميللي ثانية. يتضمن كلٌّ من تلك القيم جميع التأخيرات التي ذكرناها سابقاً، بما في ذلك تأخيرالإرسال وتأخير الانتقال وتأخير المعالجة بواسطة الموجّه وتأخير الانتظار في الصف. لمّا كان تأخير الانتظار يتفاوت مع الوقت، فإن قيمة التأخير في المحاولات الثلاث قد تختلف، بل إن تأخير رحلة الذهاب والإياب للرزمة n المرسلة إلى الموجّه n قد يتجاوز أحياناً ذلك التأخير للرزمة $n + 1$ المرسلة إلى الموجّه $n + 1$. وفي الواقع، فإننا نلاحظ هذه الظاهرة في المثال أعلاه، فقيم التأخيرات إلى الموجّه 6 أكبر منها في حالة الموجّه 7.

هل تريد الآن تجربة برنامج تتبع المسار (traceroute) بنفسك؟ نوصي بشدة بزيارة الموقع http://www.traceroute.org حيث توجد قائمة شاملة لعدد

من الروابط لواجهات ويب تتضمن مصادر لبرامج تتبع المسار. من هنا يمكنك اختيار اسم المصدر فتظهر صفحة ويب يمكنك من خلالها إدخال اسم المضيف الذي يمثل الوجهة النهائية المطلوبة، فيقوم برنامج تتبع المسار بعد ذلك (والذى يجرى تشغيله على المصدر المختار) بعمل كل شيء. هناك أيضاً عدد من البرامج المجانية لتتبع المسار توفر واجهة رسومية للمستخدِم (GUI)، منها برنامج Ping Plotter، وهو أحد البرامج المفضلة لدينا في هذا الصدد [Ping Plotter 2007].

تأخيرات الأنظمة الطرفية والتطبيقات، والتأخيرات الأخرى

بالإضافة إلى تأخيرات المعالجة والإرسال والانتقال قد توجد تأخيرات أخرى هامّة في الأنظمة الطرفية. على سبيل المثال تتسبب المودمات الهاتفية في تأخيرات تضمين (modulation) وتكويد (encoding) في حدود العشرات من الميللي ثانية. (يعتبر هذا التأخير في حالة تقنيات الوصول الأخرى كالإيثرنت ومودم الكبل، ووصلة DSL أقل أهمية وغالباً ما يهمل). عندما يرغب نظام طرفي في إرسال رزمة إلى وسط مشترك (كما في حالة شبكة WiFi أو إيثرنت) قد يضطر لتأخير الإرسال "بقصد" ضمن فعاليات بروتوكول الاشتراك في الوسط مع أنظمة طرفية أخرى. سنتناول تلك البروتوكولات بالتفصيل في الفصل الخامس.

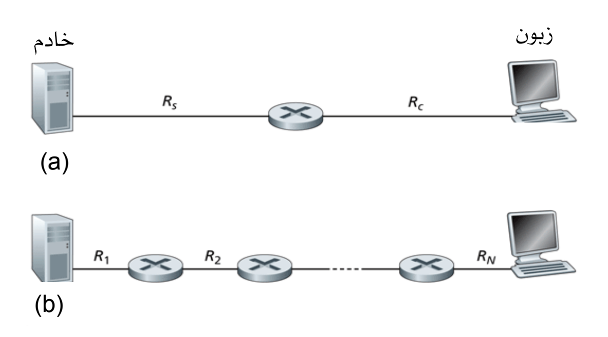
من أنواع التأخيرات الهامة الأخرى تأخير تحويل بيانات الوسائط المتعددة إلى رزم، كما هو الحال في تطبيقات إرسال مكالمات الهاتف على الإنترنت (باستخدام VoIP). في هذه التقنية ينبغي أن يملأ جانب الإرسال أولاً رزمة بالكلام الرقمي المشفر قبل إرسال الرزمة على الإنترنت، ويسمى الوقت اللازم لملء رزمة تأخير تكوين الرزم، والذي يمكن أن يكون هاماً بحيث يؤثر على جودة المكالمة الهاتفية من نوع VoIP. سنتقصى هذه القضية بشكلٍ أكبر كواجب منزلي من خلال تمرين في نهاية هذا الفصل.

1-4-4 الطاقة الإنتاجية في شبكات الحاسب

بالإضافة إلى التأخير وفقد الرزم هناك مقياس هام آخر لأداء شبكات الحاسب، ألا وهو الطاقة الإنتاجية للشبكة من طرف إلى طرف. ولتعريف تلك الطاقة الإنتاجية لنأخذ بعين الاعتبار عملية نقل ملف كبير من المضيف A إلى المضيف B عبر شبكة حاسب. قد يكون هذا الملف عبارة عن لقطة فيديو كبيرة منقولة من نظير إلى آخر ضمن نظام لمشاركة الملفات بين النظائر. تُعرّف الطاقة الإنتاجية الآنية في أي لحظة بالمعدل (بت/ثانية) الذي يتسلم به المضيف B بيانات الملف (ولعلك لاحظت من قبل أن العديد من التطبيقات، بما في ذلك الكثير من أنظمة مشاركة الملفات بين النظائر، تقوم بعرض الطاقة الإنتاجية الآنية أثناء تنزيل الملف ليقرأها المستخدم). إذا كان الملف يتألف من F بت، وتستغرق عملية النقل T ثانية، فإن الطاقة الإنتاجية المتوسطة لعملية نقل الملف تكون $\frac{F}{T}$ بت/ثانية. تحبّذ بعض التطبيقات، كإرسال مكالمات الهاتف على الإنترنت، الحصول على تأخير قليل وطاقة إنتاجية آنية تزيد بشكلٍ ثابت عن حدٍّ أدنى معين (على سبيل المثال أكثر من 24 كيلوبت/ثانية لبعض تطبيقات الهاتف على الإنترنت وأكثر من 256 كيلوبت/ثانية لبعض تطبيقات الفيديو الفورية). ولبعض التطبيقات الأخرى، بما في ذلك نقل الملفات، لا يعتبر التأخير مهماً بشكلٍ حرج، لكن يكون من المرغوب فيه الحصول على أعلى طاقة إنتاجية ممكنة.

ولإدراك ذلك المفهوم المهم للطاقة الإنتاجية بشكلٍ أفضل، دعنا نستعرض هنا بعض الأمثلة. يبين الشكل 1-15 (a) اثنين من الأنظمة الطرفية (خادم وزبون) يربط بينهما وصلتا اتصال وموجّه. لنأخذ في الاعتبار الطاقة الإنتاجية لإرسال ملف من الخادم إلى الزبون. دع R_s تمثل معدل إرسال البيانات على الوصلة بين الخادم والموجّه، وR_c تمثل معدل الإرسال على الوصلة بين الموجّه والزبون. افترض أن البتات الوحيدة التي يجري إرسالها على الشبكة ككل هي تلك البتات من الخادم إلى الزبون. نتساءل الآن ما الطاقة الإنتاجية من الخادم إلى الزبون في هذا السيناريو المثالي؟ للإجابة عن هذا السؤال يمكننا أن نتصور بتات البيانات

كسائل ووصلات الاتصال كأنابيب. واضح أن الخادم لا يستطيع ضخ البتات خلال وصلته بمعدل أسرع من R_s بت/ثانية، وكذلك لا يستطيع الموجّه توجيه البتات بمعدل أسرع من R_c بت/ثانية. إذا كانت $R_s < R_c$ فإن البتات التي يضخها الخادم سوف تتدفق إلى الزبون بمعدل R_s بت/ثانية لتعطي طاقة إنتاجية قدرها R_s بت/ثانية.

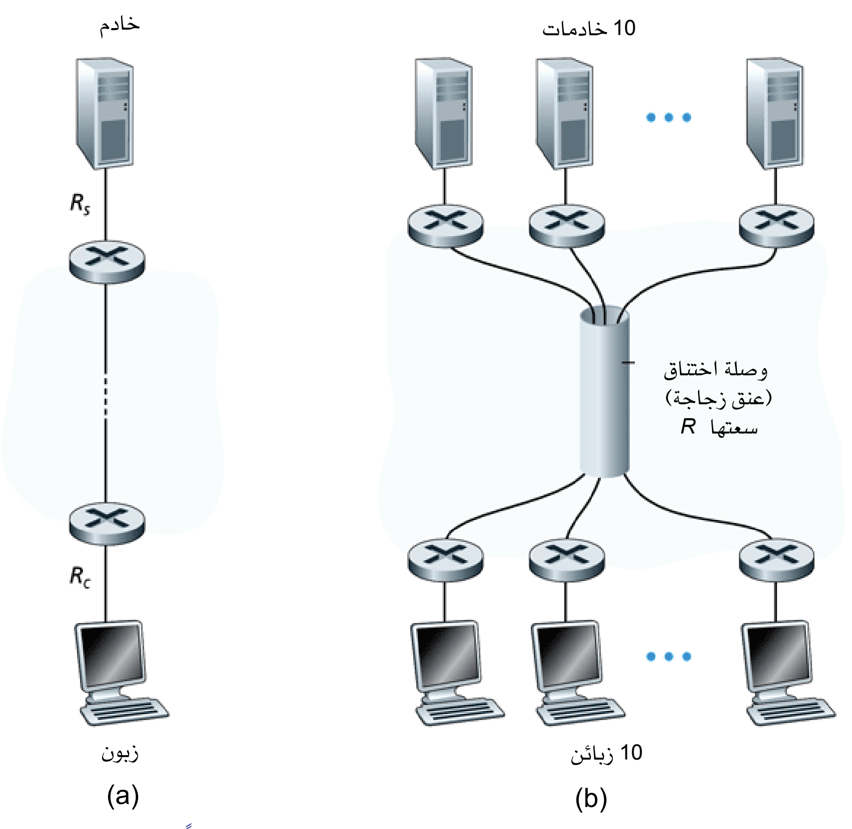


الشكل 1-15 الطاقة الإنتاجية لنقل ملف من خادم إلى زبون.

من ناحية أخرى إذا كانت $R_c < R_s$ فإن الموجّه لن يكون قادراً على إرسال البتات بنفس السرعة التي يستقبلها بها. في هذه الحالة ستترك البتات الموجّه بمعدل R_c فقط، مما يعطي طاقة إنتاجية من طرف إلى طرف قدرها R_c. (لاحظ أيضاً أنه إذا استمرت البتات في الوصول إلى الموجّه بالمعدل R_s ومغادرة الموجّه بالمعدل R_c فإن تراكم البتات التي تنتظر الإرسال إلى الزبون في الموجّه سينمو باضطراد مع الوقت ـ وهو وضع غير مرغوب فيه على الإطلاق!). وعليه فإنه في هذه الحالة البسيطة لشبكة من وصلتين، تكون الطاقة الإنتاجية هي الحد الأدنى للقيمتين (R_s, R_c)، أي $\min(R_s, R_c)$، بمعنى أنها تساوي معدل الإرسال للوصلة التي تمثل "عنق الزجاجة" على المسار من المصدر إلى الوجهة. بعد أن حسبنا الطاقة الإنتاجية، يمكننا الآن حساب الوقت التقريبي اللازم لنقل ملف كبير يتألف من F بت من الخادم إلى الزبون كـ $\frac{F}{\min(R_s, R_c)}$. كمثال عددي افترض أنك تقوم بتنزيل ملف MP3 حجمه F = 32 مليون بت، وأن الخادم له معدل

إرسال قدره R_s = 2 ميجابت/ثانية، وسرعة وصلة الوصول R_c = 1 ميجابت/ثانية، وبالتالي يكون الوقت اللازم لتنزيل الملف هو 32 ثانية. بالطبع تعتبر هذه المعادلات لحساب وقت النقل والطاقة الإنتاجية تقريبية، فهي لا تتضمن الاعتبارات الخاصة على مستوى الرزمة والبروتوكولات المستخدمة.

يبين الشكل 1-15 (b) شبكة تتكون من N وصلة بين الخادم والزبون، بمعدلات إرسال $R_1, R_2, ..., R_N$ على الوصلات المختلفة. باستخدام نفس أسلوب التحليل السابق للمثال الخاص بالشبكة ذات الوصلتين، نجد أن الطاقة الإنتاجية لإرسال الملفات من الخادم إلى الزبون هي $\min(R_1, R_2, ..., R_N)$، والتي هي مرة أخرى معدل الإرسال على وصلة عنق الزجاجة على طول المسار بين الخادم والزبون.



الشكل 1-16 الطاقة الإنتاجية من طرف إلى طرف: (a) زبون ينزّل ملفاً من خادم؛ (b) 10 زبائن ينزّلون ملفات من 10 خادمات.

لنأخذ الآن مثالاً آخر من وحي إنترنت اليوم. يبين الشكل 1-16 (a) نظامين طرفيين لخادم وزبون موصلين عبر شبكة حاسب. لنأخذ في الاعتبار الطاقة الإنتاجية لإرسال ملف من الخادم إلى الزبون. يوصّل الخادم إلى الشبكة بوصلة سرعتها R_s وكذلك يوصّل الزبون إلى الشبكة بوصلة وصول سرعتها R_c. نفترض الآن أن كل الوصلات في قلب الشبكة لها معدلات إرسال عالية جداً – أعلى بكثير من كلٍّ من R_s وR_c. في الواقع فإن قلب شبكة الإنترنت اليوم مزود بما فيه الكفاية بوصلات عالية السرعة بحيث تعاني القليل من الازدحام [Akella 2003]. افترض أيضاً أن البتات الوحيدة التي يجري إرسالها على الشبكة ككل هي تلك البتات من الخادم إلى الزبون. ونظراً لأن قلب شبكة الحاسب يشبه في هذه الحالة أنبوباً عريضاً، فإن معدل تدفق البتات من المصدر إلى الوجهة هو مرة أخرى الحد الأدنى لـ R_s وR_c، أي أن الطاقة الإنتاجية = $\min(R_s, R_c)$. لذا فإن العامل الذي يحد من الطاقة الإنتاجية للإنترنت اليوم يكمن في الغالب في شبكات الوصول.

كمثال أخير انظر الشكل 1-16 (b) الذي يمثل 10 خادمات و10 زبائن موصلين عبر قلب شبكة حاسب. في هذا المثال، هناك 10 عمليات تنزيل ملفات تتم في نفس الوقت مابين 10 خادمات و10 زبائن. لنفترض أن هذه العمليات العشر تمثل حركة مرور البيانات الوحيدة على الشبكة في ذلك الوقت. كما هو مبين في الشكل، هناك وصلة في قلب الشبكة تعبرها كل حركات المرور للعمليات العشر. دع R تمثل معدل الإرسال على تلك الوصلة، وافترض أن كل وصلات الوصول للخادمات لها نفس معدل الإرسال R_s، وكل وصلات الوصول للزبائن لها نفس معدل الإرسال R_c، وأن معدلات الإرسال على كل وصلات قلب الشبكة ماعدا الوصـــلة الواحدة المشتركة ذات المعدل R لها معدلات إرسال أكبر بكثير من كلٍّ من R_s وR_c وR. والسؤال الآن ما الطاقات الإنتاجية لعمليات التنزيل؟. واضح أنه إذا كانت سرعة الوصلة المشتركة R أكبر بكثير (مثلاً حوالي مائة مرة) من كلٍّ من R_s وR_c فإن الطاقة الإنتاجية لكل عملية تنزيل ستكون مرة أخرى $\min(R_s, R_c)$. لكن ماذا لو أن معدل إرسال الوصلة المشتركة في نفس حدود R_s وR_c؟ ماذا ستكون الطاقة الإنتاجية في هذه الحالة؟ لنلق نظرة على مثالٍ

بعينه. افترض أن R_s = 2 ميجابت/ثانية، R_c = 1 ميجابت/ثانية، R = 5 ميجابت/ثانية، وأن الوصلة المشتركة تقسّم معدل إرسالها بالتساوي بين عمليات التنزيل العشر. عندئذٍ لن يكمن عنق الزجاجة لكل عملية تنزيل في شبكة الوصول، لكنه بدلا من ذلك يصبح الآن في الوصلة المشتركة بقلب الشبكة، والتي توفر معدل إرسال قدره 500 كيلوبت/ثانية فقط كطاقة إنتاجية لكل عملية تنزيل. أي أن الطاقة الإنتاجية من طرف إلى طرف لكل عملية تنزيل تنخفض في هذه الحالة إلى 500 كيلوبت/ثانية.

توضح الأمثلة السابقة في الأشكال 1-15 و1-16 (a) أن الطاقة الإنتاجية تعتمد على معدلات الإرسال على الوصلات التي تتدفق البيانات عبرها. رأينا أنه في حالة عدم وجود تدخّل من حركة مرور أخرى على الشبكة فإن الطاقة الإنتاجية يمكن تقريبها ببساطة كمعدل الإرسال الأدنى على طول المسار بين المصدر والوجهة النهائية. أما المثال في الشكل 1-16 (b) فيوضح بشكلٍ أكثر عمومية أن الطاقة الإنتاجية لا تعتمد فقط على معدلات إرسال الوصلات على طول المسار، ولكن أيضاً على المرور المتدخّل. وبشكلٍ خاص فإن وصلة ذات معدل إرسال عالٍ قد تصبح مع ذلك عنق الزجاجة لإرسال ملفات إذا كان الكثير من البيانات الأخرى يتدفق أيضاً عبر تلك الوصلة. سنتناول الطاقة الإنتاجية في شبكات الحاسب بتفصيل أكثر في تمارين الواجب المنزلي وفي الفصول اللاحقة.

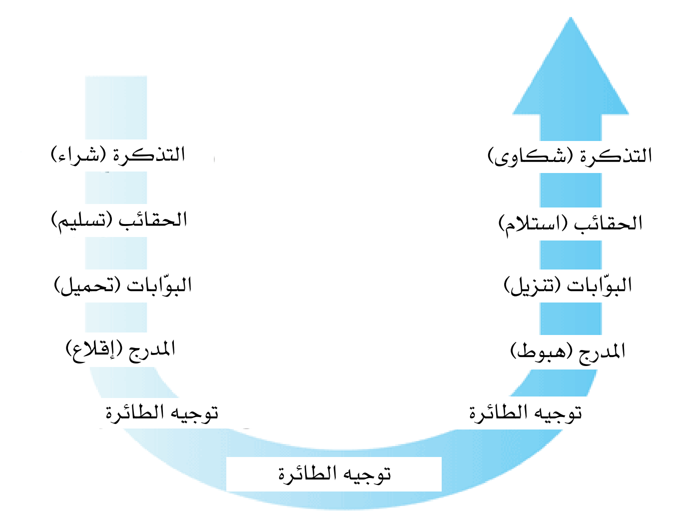
1-5 طبقات البروتوكولات ونماذج الخدمة الخاصة بها

يتضح من مناقشتنا حتى الآن أن الإنترنت نظام معقد جداً. رأينا أن هناك العديد من مكونات الإنترنت: العديد من التطبيقات والبروتوكولات، وأنواع مختلفة من الأنظمة الطرفية، ومحوّلات الرزم، وأنواع مختلفة من الأوساط المادية التي تتكون منها وصلات الاتصال. وبالنظر إلى هذا التعقيد الهائل، نتساءل: هل هناك أي أمل في تنظيم البنية المعمارية للشبكة، أو على الأقل هل يمكننا مناقشة تلك البنية؟ لحسن الحظ الإجابة على كلا السؤالين هي نعم.

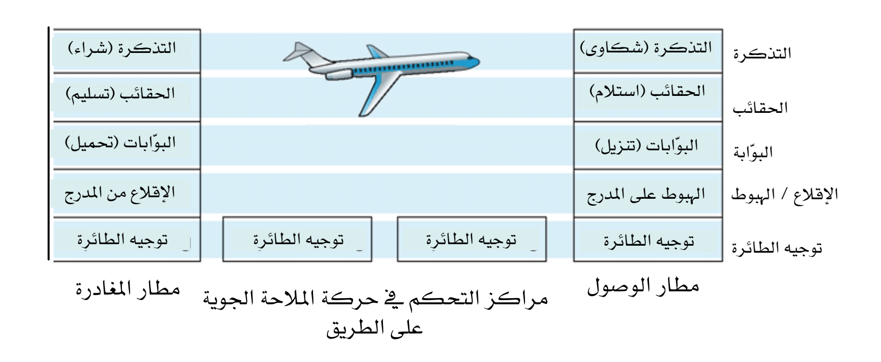
1-5-1 البنية المعمارية الطبقية

قبل محاولة تنظيم أفكارنا حول بنية الإنترنت، دعنا نبحث عن مثال بشري يقرب ذلك المفهوم. في الواقع نحن نتعامل مع أنظمة معقدة دائماً في حياتنا اليومية. تخيل أن شخصاً ما طلب منك وصف النظام الذي تقوم عليه شركات الطيران على سبيل المثال. كيف تجد التركيبة المناسبة لوصف هذا النظام المعقد الذي يتضمن وكلاء سفريات، ومدقّقي أمتعة، وموظفي بوابة، وطيارين، وطائرات، ونظاماً عالمياً للمراقبة والتحكم فى الملاحة الجوية للطائرات؟ قد تكون إحدى الطرق لوصف هذا النظام هي وصف سلسلة الخطوات التي تتخذها أنت كمسافر (أو يؤديها الآخرون لك) عندما تطير على إحدى شركات الطيران: تشتري تذكرتك، وتسلّم حقائبك، وتذهب إلى البوابة، وأخيراً تصعد على متن الطائرة. تقلع الطائرة ويتم توجيهها لتصل إلى وجهتها، وبعد أن تهبط طائرتك، تنزل منها وتأخذ حقائبك. إذا كانت الرحلة سيئة قد تشتكي إلى وكيل سفرياتك (دون جدوى!). يوضح الشكل 1-17 هذا السيناريو.

من البداية يمكننا أن نلحظ هنا بعض أوجه الشبه مع شبكات الحاسب: فأنت تُنقل من المصدر إلى الوجهة النهائية بشركة الطيران، ورزمة البيانات تُنقل من مضيف المصدر إلى مضيف الوجهة النهائية على الإنترنت. ومع ذلك فليس هذا فقط هو التناظر الذي نرمي إليه بالضبط. إننا نبحث عن شيء من الهيكلة كما يظهر فى الشكل 1-17. بالنظر إلى الشكل 1-17 نلاحظ وجود وظيفة إصدار تذاكر في كل ناحية، وهناك أيضاً وظيفة مناولة الأمتعة للمسافرين الذين لديهم تذاكر، ووظيفة بوابة للمسافرين الذين لديهم تذاكر وسلّموا أمتعتهم. أما بالنسبة للمسافرين الذين مروا خلال البوابة (أي المسافرين الذين لديهم تذاكر وسلّموا أمتعتهم وعبروا البوابة) فهناك وظائف إقلاع وهبوط. وأثناء الطيران هناك وظائف ملاحة جوية لتوجيه الطائرة. هذا يعني أنه يمكننا أن ننظر إلى الوظائف المختلفة في الشكل 1-17 بطريقة أفقية، كما هو موضح في الشكل 1-18.



الشكل 1-17 خطوات القيام برحلة بالطائرة.



الشكل 1-18 تمثيل المهام المختلفة للطيران المدني على شكل طبقات أفقية.

في الشكل 1-18 تم تقسيم وظائف شركات الطيران إلى طبقات، مما يوفر إطاراً يمكننا من خلاله مناقشة السفر على شركات الطيران. لاحظ أن كل طبقة، بالاشتراك مع الطبقات التي تحتها، تؤدي وظيفةً معينة، أي خدمة ما. فمثلاً في طبقة التذاكر وماتحتها، يتم نقل الشخص من خلال كاونتر شركة الطيران في جهة المغادرة إلى كاونتر شركة الطيران في جهة الوصول. وفي طبقة الأمتعة وما تحتها، يتم نقل أمتعة الراكب عبر تسليم واستلام الأمتعة. لاحظ أن

طبقة الأمتعة توفر هذه الخدمة فقط لشخص يحمل تذكرة. أما في طبقة البوابة فيتم نقل الراكب وأمتعته عبر بوابة مغادرة إلى بوابة وصول. وفي طبقة الإقلاع/الهبوط يتم نقل الركاب وأمتعتهم عبر مدارج إقلاع وهبوط الطائرات. تقوم كل طبقة بتوفير خدماتها عن طريق: (1) أداء بعض الأعمال ضمن تلك الطبقة (مثلاً في طبقة البوابة، تحميل وإنزال ركاب الطائرة)، (2) الاستعانة بخدمات الطبقة التي تحتها مباشرة (على سبيل المثال، تستعمل طبقة البوابة خدمة المدرج إلى المدرج من طبقة الإقلاع/الهبوط تحتها).

من مزايا البنية المعمارية على شكل طبقات أنها تمكننا من دراسة جزء محدد وواضح المعالم من نظام كبير ومعقد. لهذا التبسيط في حد ذاته قيمة كبيرة، حيث يوفر نظرة معيارية قطاعية لتصميم النظام تسهّل كثيراً تغيير طريقة إنجاز الخدمة التي توفرها طبقة ما. فطالما أن الطبقة توفر نفس الخدمات للطبقة الأعلى منها وتستعمل نفس الخدمات من الطبقة الأسفل منها، تبقى بقية النظام دون تغيير حتى عندما تتغير الطريقة المتبعة لإنجاز وظائف الطبقة. (لاحظ أن تغيير طريقة إنجاز خدمة ما هو أمر مختلف جداً عن تغيير الخدمة نفسها). على سبيل المثال، إذا تم تغيير وظائف البوابة (على سبيل المثال بجعل الركاب يستقلّون الطائرة وينزلون منها حسب الطول)، ستبقى الأجزاء الأخرى من نظام شركة الطيران بدون تغيير نظراً لأن طبقة البوابة ما زالت تؤدي نفس الوظيفة (تحميل وتنزيل الركاب) ولكنها تنفذها بطريقة مختلفة. في الأنظمة الكبيرة والمعقّدة التي يتم تحديثها باستمرار، تعتبر القدرة على تغيير طريقة إنجاز خدمة ما دون التأثير على المكونات الأخرى للنظام فائدة مهمة أخرى من فوائد استخدام البنية الطبقية.

طبقات البروتوكولات

نكتفي بهذا القدر من الحديث عن شركات الطيران لنعود إلى موضوعنا الأساسي المتعلق ببروتوكولات الشبكة. لتحقيق هيكلة تُتّبع في تصميم وظائف الشبكات، ينظم مصممو بروتوكولات الشبكة، وكذلك أجهزة وبرمجيات

الشبكة التي تنفذ تلك البروتوكولات، على شكل طبقات. ينتمي كل بروتوكول لطبقة من الطبقات، تماماً كما كانت كل وظيفة ضمن بنية شركات الطيران في الشكل 1-18 تتبع طبقة من الطبقات. مرةً أخرى سنهتم بالخدمات التي توفرها طبقة ما للطبقات فوقها ـ وهو ما يسمى بنموذج الخدمة للطبقة. كما في مثال شركات الطيران، توفر كل طبقة الخدمة المطلوبة منها عن طريق: (1) تأدية بعض الأعمال ضمن تلك الطبقة، و(2) استعمال خدمات الطبقة التي تحتها مباشرةً. على سبيل المثال قد تتضمن الخدمات التي توفرها الطبقة n تسليماً موثوقاً للرسائل من إحدى حواف الشبكة إلى الحافة الأخرى. قد ينفّذ ذلك باستعمال خدمة للتسليم غير الموثوق للرسائل من حافة إلى حافة في الطبقة n - 1 وإضافة وظيفة في الطبقة n لاكتشاف الرسائل التي فُقدت وإعادة إرسالها.

يمكن تحقيق بروتوكول طبقةٍ ما باستخدام العتاد أو البرمجيات أو مزيج منهما. ينفّذ بروتوكول طبقة التطبيقات، مثل HTTP وSMTP غالباً على شكل برمجيات في الأنظمة الطرفية، وكذلك بروتوكول طبقة النقل (Transport Layer). لكن نظراً لأن الطبقة المادية (Physical Layer) وطبقة ربط البيانات (Data Link Layer) مسؤولتان عن معالجة الاتصال على وصلة معينة، فغالباً ما ينفّذان في كروت التوصيل بالشبكة الخاصة بالوصلة المستخدمة (على سبيل المثال كروت وصلة إيثرنت أو وصلة WiFi). أما طبقة الشبكة (Network Layer) فغالباً ما تتضمن خليطاً من العتاد والبرمجيات. لاحظ أيضاً أنه كما تم توزيع الوظائف في البنية الطبقية لشركات الطيران بين المطارات المختلفة ومراكز التحكم في الطيران التى يتكون منها النظام، تم أيضاً توزيع بروتوكول الطبقة n بين الأنظمة الطرفية ومحوّلات الرزم وغيرها من المكونات الأخرى التي تتألف منها الشبكة، بمعنى أنه يوجد في أغلب الأحيان جزء من بروتوكول الطبقة n في كل من مكونات الشبكة هذه.

لطبقات البروتوكول فوائد من حيث المفهوم والهيكل. فكما رأينا، يوفراستخدام الطبقات طريقة هيكلية منظمة لمناقشة مكونات النظام. كما تؤدي المعيارية والقطاعية الناتجة إلى تسهيل تحديث مكونات النظام. ومع ذلك

فإن بعض المهندسين والباحثين في مجال الشبكات يعارضون بشدة طريقة الطبقات [Wakeman 1992]. فمن العيوب المحتملة لنظام الطبقات الازدواجية وذلك بتكرار طبقة ما لوظيفة طبقة أخرى. فعلى سبيل المثال، توفر العديد من طبقات البروتوكولات المعيارية وظائف تصحيح الأخطاء الناجمة عن الإرسال على مستوى كل وصلة (في طبقة ربط البيانات) وأيضاً على المستوى من طرف إلى طرف (في طبقة النقل). ومن العيوب المحتملة أيضاً أنه للقيام بوظيفتها، قد تحتاج طبقة من الطبقات إلى معلومات لا تتوافر إلا في طبقة أخرى (على سبيل المثال خاتم بقيمة الوقت الحالي (time stamp))، مما يتنافى مع هدف الفصل بين الطبقات.

لاحظ أنه عند الحديث عن طبقات البروتوكولات مجتمعة، يطلق عليها رصة البروتوكولات (protocol stack). تتضمن رصة بروتوكولات الإنترنت خمس طبقات: المادية، وربط البيانات، والشبكة، والنقل، والتطبيقات، كما هو موضح فى الشكل 1-19 (a). وإذا تفحصت فهرس محتويات هذا الكتاب، فسترى أننا رتبناه تقريباً على أساس طبقات رصة بروتوكولات الإنترنت متبعين أسلوب من أعلى إلى أسفل، حيث نتناول أولاً طبقة التطبيقات ومن ثم نتجه لأسفل لدراسة الطبقات الأخرى التي تقع تحتها.

(b) نموذج OSI المرجعي بسبع طبقات
التطبيقات
التقديم
الجلسة
النقل
الشبكة
ربط البيانات
المادية

(a) رصّة بروتوكولات الإنترنت بخمس طبقات
التطبيقات
النقل
الشبكة
ربط البيانات
المادية

الشكل 1-19 (a) رصة بروتوكول الإنترنت، (b) نموذج OSI المرجعي.

طبقة التطبيقات

طبقة التطبيقات هي الطبقة التي تضم تطبيقات الشبكة وبروتوكولات طبقة التطبيقات. تتضمّن طبقة التطبيقات في شبكة الإنترنت العديد من البروتوكولات، كبروتوكول HTTP (الذي يوفر خدمات طلب ونقل وثيقة على الويب)، وبروتوكول SMTP (الذي يوفر خدمات نقل رسائل البريد الإلكتروني)، وبروتوكول FTP (الذي يوفر خدمات نقل الملفات بين نظامين طرفيين). سنرى أيضاً أن وظائف معينة لطبقة الشبكة يتم تشغيلها بمعاونة بروتوكولات فى طبقة التطبيقات مثل بروتوكول DNS (نظام أسماء النطاقات) والذى يحوّل ما بين اسم النطاق (تلك الصيغة المناسبة للتعامل البشري كـ www.ietf.org) وعنوان الـ IP الرقمي المناظر له على الشبكة بطول 32 بتاً (في حالة بروتوكول الإنترنت من الإصدار الرابع (IPv4)). سنرى في الفصل الثانى أنه من السهل جداً تطوير واستعمال بروتوكولات جديدة خاصة بنا لطبقة التطبيقات.

يوزّع بروتوكول طبقة التطبيقات على العديد من الأنظمة الطرفية، حيث يستخدمه التطبيق على نظام طرفي ما في تبادل رزم البيانات مع التطبيق الموجود على نظام طرفي آخر. سنطلق على رزمة المعلومات هذه في طبقة التطبيقات اسم رسالة (message).

طبقة النقل

تقوم طبقة النقل بالإنترنت بنقل رسائل طبقة التطبيقات بين الأنظمة الطرفية على الشبكة. ويستخدم على الإنترنت بروتوكولان لنقل البيانات هما بروتوكول التحكم في الإرسال (TCP) وبروتوكول وحدة بيانات المستخدِم (UDP)، حيث يمكن لأي منهما نقل رسائل طبقة التطبيقات. يوفر بروتوكول TCP خدمة توصيلية (connection-oriented service) للتطبيقات التي تستخدمه، وتتضمن هذه الخدمة تسليماً مضموناً لرسائل طبقة التطبيقات إلى وجهتها النهائية وضبطاً لمعدل تدفق البيانات (بمعنى تعديل سرعة المرسل لتلائم سرعة المستقبل). كما يقوم بروتوكول TCP أيضاً بتجزئة الرسائل الطويلة إلى قطع أقصر، وتوفير آلية للسيطرة على الازدحام (congestion control) تقوم بالحدِّ من

معدل الإرسال عندما تكون الشبكة مزدحمة. أما بروتوكول UDP فيوفر خدمة لاتوصيلية (connectionless service) لتطبيقاته. هذه الخدمة بسيطة فهي لا توفر أي اعتمادية أو ضبط لمعدل التدفق أو سيطرة على الازدحام. في هذا الكتاب سنطلق على رزمة البيانات في طبقة النقل اسم قطعة (segment).

طبقة الشبكة

طبقة الشبكة في الإنترنت مسؤولة عن نقل حزم البيانات في طبقة الشبكة، والتي تُعرف بوحدة البيانات (datagram) أو رزمة (packet)، من مضيف إلى آخر. يقوم بروتوكول طبقة النقل في الإنترنت (TCP أو UDP) في مضيف بدفع قطعة بيانات طبقة النقل متضمنةً عنوان وجهتها النهائية إلى طبقة الشبكة، تماماً كما تقوم أنت بتسليم خطاب يحمل عنوان المرسل إليه إلى خدمة البريد. بعد ذلك تقوم طبقة الشبكة في مضيف الوجهة النهائية بتوفير خدمة تسليم القطعة إلى طبقة النقل عند وصولها.

تتضمن طبقة شبكة الإنترنت بروتوكول الإنترنت الشهير IP، والذي يعرّف الحقول المختلفة في وحدة البيانات كما يحدد كيفية تصرّف الأنظمة الطرفية والموجّهات إزاء تلك الحقول. ويوجد بروتوكول IP وحيد لطبقة شبكة الإنترنت، وعلى كل مكونات الإنترنت التي لها طبقة شبكة أن تنفذ هذا البروتوكول. كما تتضمن طبقة شبكة الإنترنت أيضاً بروتوكولات التوجيه التي تحدد المسارات التي تسلكها وحدات البيانات بين المصادر والوجهات النهائية. يتوافر العديد من بروتوكولات التوجيه على الإنترنت. كما رأينا في الجزء 1-3 فالإنترنت هي شبكة تتألف من شبكات، وبوسع المشرف على كل شبكة استخدام أي بروتوكول توجيه يرغب فيه. رغم أن طبقة الشبكة تحتوي على العديد من بروتوكولات التوجيه بجانب البروتوكول IP، فإن هذه الطبقة يطلق عليها غالباً ببساطة طبقة IP، مما يعكس حقيقة أن هذا البروتوكول يُعد بمثابة الصمغ الذي يربط الإنترنت ببعضها البعض.

طبقة ربط البيانات

تقوم طبقة شبكة الإنترنت بتوجيه وحدة البيانات (datagram) عبر سلسلة من الموجّهات بين المصدر والوجهة النهائية. ولنقل وحدة البيانات من عقدة على الشبكة (مضيف أو موجّه) إلى العقدة التالية على طول المسار، تعتمد طبقة الشبكة على خدمات طبقة ربط البيانات. وبالتحديد تقوم طبقة الشبكة في كل عقدة بدفع وحدة البيانات إلى طبقة ربط البيانات أسفلها، والتي تقوم بدورها بتسليم وحدة البيانات إلى العقدة التالية على المسار. في تلك العقدة التالية تقوم طبقة ربط البيانات برفع وحدة البيانات الواصلة إلى طبقة الشبكة أعلاها.

تعتمد الخدمات التي توفرها طبقة ربط البيانات على بروتوكول طبقة ربط البيانات المستخدم على الوصلة. على سبيل المثال توفر بعض بروتوكولات طبقة ربط البيانات تسليماً موثوقاً من عقدة الإرسال إلى عقدة الاستلام عبر وصلة واحدة. لاحظ أن خدمة التسليم الموثوق هذه تختلف عن خدمة التسليم الموثوق لبروتوكول التحكم في الإرسال TCP (والمستخدم فى طبقة النقل) والتي توفر تسليماً موثوقاً من نظام طرفي إلى نظام طرفي آخر. من أمثلة بروتوكولات طبقة ربط البيانات بروتوكول إيثرنت، وبروتوكول WiFi للاتصال اللاسلكى، وبروتوكول التوصيل من نقطة إلى نقطة PPP. ونظراً لأن وحدات البيانات (datagrams) تحتاج عادةً لعبور عدة وصلات للانتقال من المصدر إلى الوجهة النهائية، فقد تعالج تلك الوحدات ببروتوكولات مختلفة في طبقات ربط البيانات في العُقد التي تمر بها على طول مسارها. على سبيل المثال يمكن لوحدة بيانات أن تعالَج ببروتوكول الإيثرنت على وصلة وببروتوكول PPP على الوصلة التي تليها. ومن ثم فإن طبقة الشبكة في العقد المختلفة تتلقى خدمات مختلفة من كلٍّ من بروتوكولات طبقة ربط البيانات المختلفة. في هذا الكتاب سنطلق على رزمة البيانات في طبقة ربط البيانات اسم إطار (frame).

الطبقة المادية

بينما تتلخص وظيفة طبقة ربط البيانات في نقل إطارات كاملة من عقدة إلى العقدة المجاورة لها على الشبكة، فإن الشغل الشاغل للطبقة المادية ينحصر في نقل البتات المفردة التي تكوّن إطار طبقة ربط البيانات من عقدة لأخرى. ومرةً أخرى تختلف البروتوكولات المستخدمة في هذه الطبقة تبعاً لنوع الوصلة وتعتمد على وسط الإرسال المادي للوصلة (على سبيل المثال زوج أسلاك نحاسية مجدولة، أو ألياف ضوئية وحيدة النمط (single mode). فمثلاً للإيثرنت العديد من بروتوكولات الطبقة المادية: واحد لزوج الأسلاك النحاسية المجدول، وآخر للكبل المحوري، وآخر لليف الضوئي. وهكذا ففي كل حالة يتم نقل البتات عبر الوصلة على نحو مختلف.

نموذج OSI لبنية الشبكات

بعد أن استعرضنا رصة بروتوكولات الإنترنت بالتفصيل، يجدر بنا أن نذكر أن هذا النموذج ليس هو الوحيد من نوعه على الساحة. وبالتحديد، في أواخر السبعينيات من القرن الماضي اقترحت المنظمة الدولية للمعايير ISO تنظيماً لشبكات الحاسب مكوناً من سبع طبقات، وأطلقت عليه نموذج ترابط الأنظمة المفتوح (Open Systems Interconnection (OSI)) [ISO 2007]. تَشكّل هذا النموذج بينما كانت بروتوكولات الإنترنت في مهدها كمجرد مجموعة من عدة أطقم من البروتوكولات المختلفة تحت التطوير. وفي الواقع من المحتمل أن مخترعي نموذج ترابط الأنظمة المفتوح الأصلي لم يكن في بالهم الإنترنت عندما قاموا بتطوير ذلك النموذج. ومع ذلك فابتداءً من أواخر السبعينيات، اعتمد العديد من المقررات التى تدرس بالجامعات ومراكز التدريب نموذج OSI ذا الطبقات السبع كمحور لمقرراتهم الدراسية ودوراتهم التدريبية. وبسبب تأثيره المبكّر على التعليم في مجال الشبكات، لا يزال نموذج OSI يفرض وجوده في بعض الكتب والدورات التدريبية عن الشبكات.

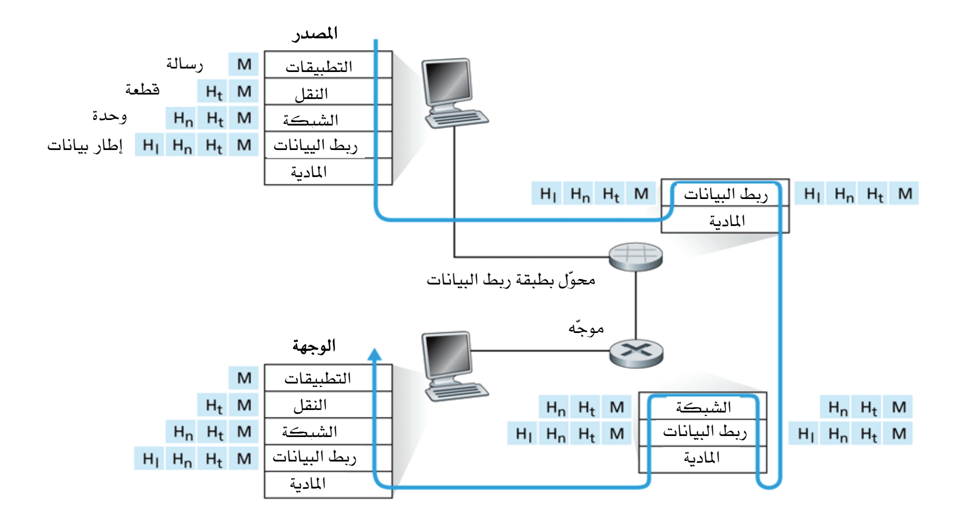
كما يوضح الشكل 1-19 (b) يضم نموذج OSI المرجعي الطبقات السبع التالية: طبقة التطبيقات، طبقة التقديم، طبقة الجلسة، طبقة النقل، طبقة الشبكة، طبقة ربط البيانات، والطبقة المادية. خمسٌ من تلك الطبقات لها تقريباً نفس وظائف الطبقات المناظرة في بروتوكولات الإنترنت. وعليه فسنتناول هنا الطبقتين الإضافيتين الموجودتين في نموذج OSI، ألا وهما طبقة التقديم وطبقة الجلسة. يتلخص دور طبقة التقديم في توفير الخدمات التي تمكّن التطبيقات المختلفة من استيعاب معنى البيانات التي يتم تبادلها عبر الاتصال. تتضمّن تلك الخدمات ضغط وتشفير البيانات، بالإضافة إلى وصف البيانات (والذي، كما سنرى في الفصل التاسع، يريح التطبيقات من عناء القلق حول الصيغة الداخلية المستخدمة لتمثيل البيانات وتخزينها، والتي قد تختلف من حاسب لآخر). أما طبقة الجلسة فتوفر وسائل لتحديد عملية تبادل البيانات وتحقيق تزامنها، بما في ذلك طرقاً لإدخال نقاط للفحص وأساليب لاستعادة البيانات عند حدوث أعطاب.

إن افتقار الإنترنت إلى طبقتين موجودتين في نموذج OSI المرجعي يطرح سؤالين مهمّين: هل الخدمات التي توفرها هاتان الطبقتان غير مهمة؟. ماذا لو احتاج أحد التطبيقات إلى تلك الخدمات؟ للإنترنت نفس الجواب عن كلا السؤالين ـ الأمر يعود لمطوّر التطبيقات، فعليه أن يقرر ما إذا كانت الخدمة مهمة، وعندئذ يُضمّنها فى التطبيق.

1-5-2 الرسائل والقطع ووحدة البيانات والإطارات

يبين الشكل 1-20 الطريق المادي الذي تسلكه البيانات من أعلى إلى أسفل عبر رصة البروتوكولات في نظام طرفي مُرسِل، ثم من أسفل إلى أعلى والعكس عبر رصات البروتوكولات على الطريق في محوّلات طبقة ربط البيانات والموجّهات، ثم في النهاية من أسفل إلى أعلى خلال رصة البروتوكولات في النظام الطرفي المستقبِل. كما سنلاحظ لاحقاً في هذا الكتاب، تعتبر محوّلات طبقة ربط البيانات والموجّهات محوّلات رزم. كما هو الحال في الأنظمة الطرفية تنظّم تلك المحوّلات والموجّهات عتاد وبرمجيات الشبكة لديها على شكل طبقات، غير أنها

لا تستخدم كل الطبقات المعروفة في رصة البروتوكولات وتكتفي بالطبقات السفلى منها فقط. فمثلاً كما هو مبين في الشكل 1-20 تقتصر محوّلات طبقة ربط البيانات على الطبقات 1 و2 في حين يقتصر الموجّه على الطبقات من 1 إلى 3. هذا يعني على سبيل المثال أن موجّهات الإنترنت قادرة على تنفيذ بروتوكول الإنترنت (بروتوكول الطبقة 3)، بينما لا تستطيع ذلك محوّلات طبقة ربط البيانات. سنرى لاحقاً أنه بينما لا تستطيع محوّلات طبقة ربط البيانات التعرف على عناوين IP، فإنه بوسعها التعرف على عناوين الطبقة 2 (كعناوين الإيثرنت). لاحظ أن المضيفات تستخدم كل الطبقات الخمس، وهذا يتسق مع وجهة النظر القائلة بأن بنية الإنترنت تركز معظم التعقيد على حافة الشبكة.



الشكل 1-20 يتضمن كلٌّ من المضيفات والموجهات ومحوّلات طبقة ربط البيانات مجموعة مختلفة من الطبقات حسب الوظائف التي يؤديها كلٌّ منها.

يوضح الشكل 1-20 مفهوماً مهماً آخر ألا وهو مفهوم تغليف البيانات (encapsulation) في الطبقات المختلفة. في مضيف الإرسال تُدفع الرسالة (M في الشكل 1-20) من طبقة التطبيقات إلى طبقة النقل. في أبسط الحالات تأخذ طبقة النقل تلك الرسالة وتلحق بها بيانات إضافية تعرف بالترويسة (header) الخاصة بطبقة النقل (H_t في الشكل 1-20) والتي ستستعمل من قِبَل طبقة النقل على ناحية المستقبِل. تشكّل رسالة طبقة التطبيقات وترويسة طبقة النقل معاً قطعة البيانات (segment) الخاصة بطبقة النقل، ومن ثم فإن قطعة طبقة النقل تغلّف رسالة طبقة التطبيقات. قد تتضمّن البيانات الإضافية التي يتم إلحاقها أثناء عملية التغليف معلومات تمكّن طبقة النقل لدى المستقبِل من تسليم الرسالة إلى التطبيق المناسب، أو بتات إضافية خاصة باكتشاف الخطأ تسمح للمستلم بتحديد ما إذا كانت بتات الرسالة المستقبلة قد طرأ عليها تغيير أثناء انتقالها في الطريق. تقوم طبقة النقل بعد ذلك بدفع قطعة البيانات إلى طبقة الشبكة، والتي تقوم بدورها بإضافة بيانات ترويسة طبقة الشبكة (H_n في الشكل 1-20) كعناوين النظامين الطرفيين اللذين يمثلان المصدر والوجهة النهائية، ومن ثم تكوّن وحدة البيانات (datagram) الخاصة بطبقة الشبكة. تُدفع وحدة البيانات إلى طبقة ربط البيانات والتي (بالطبع!) ستضيف بيانات الترويسة الخاصة بها لتكوين إطار (frame) طبقة ربط البيانات. وهكذا نرى أن رزمة البيانات في كل طبقة تتألف من نوعين من الحقول: حقل الترويسة الخاصة بتلك الطبقة وحقل البيانات (الحمل الآجر (payload)) (والذى يمثّل رزمة البيانات الواصلة إليها من الطبقة الأعلى).

هناك تناظر مفيد هنا يتمثل في إرسال مذكرة داخلية من مكتب أحد فروع شركة إلى مكتب فرع آخر عن طريق خدمة البريد العمومي، فالمذكرة تناظر رسالة طبقة التطبيقات. افترض أن أليس التي تعمل في مكتب الفرع الأول تريد إرسال مذكرة إلى بوب في مكتب الفرع الآخر. تضع أليس المذكرة في المظروف الخاص بالبريد الداخلي وتضع اسم بوب ورقم القسم الذي يعمل به على المظروف. إن مظروف البريد الداخلي بمحتوياته يناظر قطعة بيانات طبقة النقل حيث يحتوي على معلومات الترويسة (اسم بوب ورقم القسم) كما أنه يغلّف رسالة طبقة التطبيقات (المذكرة). وعندما تتلقى غرفة بريد مكتب الفرع

المُرسِل مظروف البريد الداخلي، يتم وضع ذلك المظروف داخل مظروف آخر مناسب للإرسال عبر خدمة البريد العمومي. تكتب غرفة البريد العنوان البريدي لكلٍّ من مكتبي فرع الإرسال والاستلام على المظروف البريدي. لاحظ هنا أن هذا المظروف البريدي يناظر وحدة البيانات لطبقة الشبكة، حيث إنه يغلّف قطعة طبقة النقل (مظروف البريد الداخلي)، والذي يغلّف بدوره الرسالة الأصلية (المذكرة). تُسلّم الخدمة البريدية المظروف البريدي إلى غرفة بريد مكتب فرع الاستلام، وهناك تبدأ عملية استخراج الرسائل (عكس عملية التغليف)، وانتزاع مظروف البريد الداخلي وتوصيله إلى بوب. وأخيراً يفتح بوب المظروف ويطّلع على المذكرة.

يمكن أن تكون عملية التغليف أكثر تعقيداً من ذلك. فعلى سبيل المثال قد يتطلب الأمر لدى المرسل تجزئة رسالة كبيرة إلى عدد من قطع طبقة النقل (والتي قد تُقسِّم كل واحدة منها إلى عدد من وحدات بيانات طبقة الشبكة). في هذه الحالة يتعين على المستقبِل إعادة بناء قطعة البيانات تلك من وحدات البيانات المكوّنة لها.

6-1 أمن الشبكات

أصبحت الإنترنت ذات أهمية كبيرة للعديد من المؤسسات اليوم، بما في ذلك الشركات الكبيرة والصغيرة، والجامعات، والمصالح الحكومية، حيث يعتمد الملايين من الأفراد على الإنترنت في العديد من نشاطاتهم الشخصية والاجتماعية والمهنية. لكن للأسف يكمن وراء كل هذه الفوائد الهامة جانب مظلم، حيث يحاول "أناس سيئون" إيقاع الخراب في حياتنا اليومية بإتلاف حاسباتنا المرتبطة بالإنترنت، وبانتهاك خصوصيتنا، وتعطيل خدمات الإنترنت التي نعتمد عليها في حياتنا [Skoudis 2006].

يتناول مجال أمن الشبكات الطرق التي يمكن أن يستخدمها هؤلاء المخربون في الهجوم على شبكات الحاسب، وكيف يمكننا نحن كخبراء المستقبل في شبكات الحاسب أن ندافع عن الشبكات ضدّ تلك الهجمات، أو

لعله يكون من الأفضل أن نصمم بنية معمارية جديدة للشبكة تكون لديها من البداية المناعة ضدّ مثل تلك الهجمات. بالنظر إلى تكرار وتنوع الهجمات حالياً، بالإضافة إلى التهديدات التدميرية المتوقعة في المستقبل، أصبح أمن الشبكات موضوعاً أساسياً في مجال شبكات الحاسب في السنوات الأخيرة. من مميزات هذه الطبعة الرابعة من هذا الكتاب إبرازها لقضايا أمن الشبكات ووضعها في المقدمة. سنبدأ حملتنا في أمن الشبكات في هذا الجزء، حيث سنستعرض باختصار بعض الهجمات السائدة والأكثر ضرراً على الإنترنت اليوم. بعد ذلك وأثناء تناولنا لتقنيات وبروتوكولات شبكات الحاسب المختلفة بتفصيل أكثر في الفصول اللاحقة، سنأخذ بعين الاعتبار القضايا المختلفة المتعلقة بالأمن والمرتبطة بتلك التقنيات والبروتوكولات. وأخيراً في الفصل الثامن وقد تسلحنا بالخبرة في شبكات الحاسب وبروتوكولات الإنترنت، سندرس بعمق كيف يمكن الدفاع عن شبكات الحاسب ضدّ الهجمات المختلفة، وكذلك تصميم تلك الشبكات وتشغيلها لجعل مثل تلك الهجمات مستحيلة الحدوث بالدرجة الأولى.

ولأن خبرتك ما زالت ضعيفة في شبكات الحاسب وبروتوكولات الإنترنت، فسنبدأ هنا فقط باستعراض بعض المشاكل المتعلقة بالأمن والمنتشرة بكثرة هذه الأيام. هذا سيحفزنا لمناقشات أوسع وأعمق في الفصول القادمة. لنبدأ هنا ببساطة بالتساؤلات التالية: ما مجالات الضرر الذي يمكن أن يحدث؟ ما مدى حساسية شبكات الحاسب للاختراق؟ ما هي بعض أنواع الهجوم على الشبكة المنتشرة اليوم؟ للإجابة على تلك التساؤلات، إليك بعض ما يمكن أن يفعله الأشرار:

وضع برمجيات خبيثة على مضيفك من خلال الإنترنت

نقوم بربط أجهزتنا بالإنترنت لأننا نريد استقبال وإرسال البيانات من الإنترنت وإليها. تتضمن تلك البيانات كل أنواع المواد الجيدة، بما في ذلك صفحات الويب ورسائل البريد الإلكتروني وملفات MP3 والمكالمات الهاتفية والفيديو الحي ونتائج محركات البحث وما إلى ذلك. لكن ولسوء الحظ مع

كل تلك المواد الجيدة تجيء أيضاً مواد خبيثة تعرف على العموم بالبرمجيات الخبيثة (malware) والتي يمكن أن تدخل أجهزتنا أيضاً وتصيبها. عندما تصيب البرامج الخبيثة أجهزتنا يمكن أن تقترف كل أنواع الأعمال المؤذية، بما في ذلك حذف ملفاتنا، وتركيب برامج تجسس تجمع معلوماتنا الخصوصية كأرقام الضمان الاجتماعي وكلمات السر وضرباتنا على لوحة المحوّلات، وترسلها (على الإنترنت طبعاً!) إلى الأشرار. بل ربّما يكون قد تم أيضاً تسجيل مضيفِنا المُختَرَق ضمن شبكة من آلاف الأجهزة المُختَرَقة بنفس الطريقة، تُعرف بشكلٍ جماعي بشبكة الروبوت (botnet)، والتى يسيطر عليها الأشرار ويستخدمونها لتوزيع رسائل بريد الدعاية الإلكتروني (spam) أو تنسيق هجمات موزعة لحجب الخدمة (Distributed Denial-of-Service (DDoS)) (والتى سنناقشها قريباً) ضدّ المضيفات المستهدفة.

معظم البرامج الخبيثة الموجودة اليوم ذاتية التكاثر والانتشار، بمعنى أنه عندما يصاب مضيف ما، تسعى تلك البرامج للانتشار من ذلك المضيف والدخول إلى مضيفات أخرى على الإنترنت، ومن تلك المضيفات المصابة مؤخراً، تسعى للدخول إلى مضيفات أخرى، وهكذا. بهذا الأسلوب يمكن للبرامج الخبيثة ذاتية التكاثر أن تنتشر بمعدل أسي سريع. على سبيل المثال، تضاعف عدد الأجهزة المصابة بدودة Slammer (والمعروفة أيضاً بـ Saphire) عام 2003 بمعدل مرة كل 8.5 ثانية في الدقائق القليلة الأولى من تفشّيها، مما أدى إلى إصابة أكثر من 90 بالمائة من المضيفات الضعيفة أمام الهجوم خلال 10 دقائق [Moore 2003]. يمكن أن تنتشر البرامج الخبيثة على شكل فيروس، أو دودة، أو حصان طروادة [Skoudis 2004]. تتطلب الفيروسات نوعاً من التفاعل من جانب المستخدم لتحقيق الإصابة لجهاز المستخدم. ويعتبر المثال التقليدي لذلك الملف المرفق مع رسالة بريد إلكتروني والذي يحتوي على برنامج خبيث قابل للتنفيذ. فإذا استلم المستخدم الرسالة وفتح ذلك الملف، سيقوم المستخدم بدون قصد بتشغيل البرنامج الخبيث على الجهاز. عادةً ما تكون فيروسات البريد الإلكتروني من هذا النوع ذاتية الاستنساخ (أى بمجرد تشغيل البرنامج، يمكن على سبيل المثال

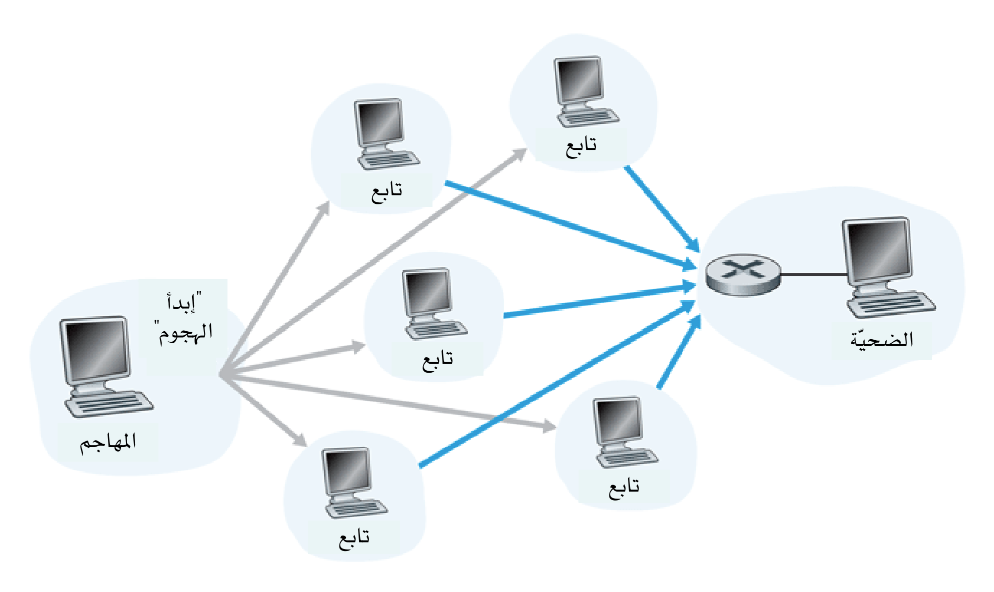
أن يرسل الفيروس رسالة مطابقة بمرفق خبيث مطابق إلى كل عنوان في دفتر عناوين المستخدِم. أما الديدان (مثل دودة Slammer) فيمكنها دخول الجهاز بدون أي تفاعل معين من جانب المستخدم. على سبيل المثال، عندما يقوم المستخدم بتشغيل تطبيق ضعيف على الشبكة يكون بوسع المهاجم إرسال برامج خبيثة إليه. في بعض الحالات وبدون أي تدخل يمكن أن يقبل التطبيق البرنامج الخبيث من الإنترنت ويشغِّله، ومن ثم تتكون دودة. تقوم الدودة في الجهاز المصاب مؤخراً بمسح الإنترنت بحثاً عن المضيفات الأخرى التى تشغّل نفس التطبيق الضعيف على الشبكة. وعندما تجد الدودة مضيفات ضعيفة أخرى فإنها ترسل نسخة من نفسها إلى تلك المضيفات. وأخيراً حصان طروادة هو برنامج خبيث يتمثل كجزء مستتر ضمن برامج مفيدة عادةً. اتسعت البرامج الخبيثة اليوم وانتشرت انتشار كبيراً أصبح يكلفنا الكثير. فعلى سبيل المثال، تم تقدير الضرر المالي للفيروسات وحدها بأكثر من 14 بليون دولار في عام 2005 [Malware 2006]. لذا بينما تمضي أيها القارئ قدماً في استيعاب هذا الكتاب، نحثك على التفكير في الإجابة على السؤال التالي: ماذا يمكن لمصممي شبكات الحاسب أن يفعلوه لحماية الأجهزة الموصلة بالإنترنت من هجمات البرامج الخبيثة؟

مهاجمة الخادمات والبنية التحتية للشبكة

هناك عدد كبير من تهديدات أمن الشبكات يمكن تصنيفها كهجمات لحجب الخدمة (DoS). كما يتضح من الاسم، يؤدي هذا النوع من الهجوم إلى جعل شبكة أو مضيف أو جزء آخر من البنية التحتية غير متاحٍ للاستعمال من قِبَل المستخدمين الشرعيين. فيمكن أن يكون كلٌّ من خادمات الويب وخادمات البريد الإلكتروني وخادمات أسماء النطاقات DNS (ستناقش في الفصل الثانى) وشبكات المؤسسات عرضةً لهجمات حجب الخدمة. وهذه الهجمات منتشرة جداً على الإنترنت، حيث تحدث آلاف الهجمات منها كل عام [Moore 2001; Mirkovic 2005]. تنضوي معظم هجمات حجب الخدمة في الإنترنت تحت واحد من الأصناف الثلاثة التالية:

- الهجوم باستهداف نقاط الضعف: يتضمن هذا الهجوم إرسال بضعة رسائل مصممة بشكلٍ جيد إلى تطبيق أو نظام تشغيل يعاني من نقاط ضعف يجري تشغيله على المضيف المستهدف. إذا تم إرسال التسلسل الصحيح من تلك الرزم إلى تطبيق أو نظام تشغيل ضعيف، فإنه يمكن إيقاف الخدمة، أو أسوأ من ذلك يمكن أن يتوقف المضيف عن العمل تماماً.
- هجوم فيضان الحيز الترددي: يرسل المهاجم فيضاناً من الرزم إلى المضيف المستهدف، بحيث يتم عرقلة وصلة الوصول الخاصة بالمضيف مما يمنع وصول الرزم الشرعية إليه.
- هجوم فيضان التوصيلات: ينشئ المهاجم عدداً كبيراً من توصيلات بروتوكول TCP مفتوحة بالكامل أو نصف مفتوحة لدى المضيف المستهدف (سنتناول توصيلات TCP في الفصل الثالث). يمكن أن يصبح المضيف محمّلاً جداً بهذه التوصيلات المزيفة بحيث يتوقف عن قبول توصيلات شرعية.

دعنا الآن نستكشف هجوم فيضان الحيز الترددي بشيء من التفصيل. من مناقشتنا لتحليل التأخير وفقد الرزم في الجزء 1-4-2، يتضح أنه إذا كان معدل وصول البيانات للخادم هو R بت/ثانية، فإن المهاجم سيحتاج لإرسال حركة مرور بمعدل R بت/ثانية تقريباً لكي يتمكن من إلحاق الضرر. فإذا كانت R كبيرة جداً، فإن مصدراً واحداً للهجوم لن يتمكن من توليد حركة مرور كافية لإيذاء الخادم. وعلاوة على ذلك إذا انبثقت حركة المرور من مصدر واحد، فإن موجّهاً في مدخل الشبكة قد يتمكن من اكتشاف الهجوم ومنع كل حركة المرور من ذلك المصدر قبل أن تقترب من الخادم. لهذا السبب يستخدم المهاجم أسلوب حجب الخدمة الموزّع (DDoS)، حيث يسيطر المهاجم على عدة مصادر (كما يتضح من الشكل 1-21) ويجعل كل مصدر يرسل سيلاً من حركة المرور إلى المضيف المستهدف. بهذه الطريقة يحتاج معدل حركة المرور الكلي عبر كل المصادر المشاركة في الهجوم أن يكون قريباً من R لشل الخدمة.



الشكل 1-21 هجوم موزّع لحجب الخدمة.

تعتبر هجمات DDoS التي تستخدم شبكات هجومية (botnets) تضم آلاف المضيفات المخترقة أمراً شائع الحدوث اليوم [Mirkovic 2005]. ويلاحظ أن هجمات حجب الخدمة الموزّعة تعتبر أصعب بكثير في اكتشافها والوقاية منها من هجوم مماثل من مضيف واحد.

ومرةً أخرى نشجعك على النظر في هذا السؤال بينما تشق طريقك عبر هذا الكتاب: ما الذي يمكن لمصممي شبكات الحاسب عمله للوقاية من هجمات حجب الخدمة؟ سنرى فيما بعد أن الأمر يحتاج إلى أساليب مختلفة من الدفاعات للتعامل مع الأنواع الثلاثة المذكورة من هجمات حجب الخدمة.

"التقاط" الرزم (Packet Sniffing)

يتواصل الكثير من مستخدمي الإنترنت اليوم مع الشبكة عن طريق أجهزة لاسلكية، كالحاسبات النقّالة الموصلة عن طريق شبكة WiFi أو الأدوات المحمولة باليد والمرتبطة عن طريق وصلات إنترنت خلوية (والتى سيتم تغطيتها في الفصل السادس). بينما يُعتبر الوصول للإنترنت من كل

مكان أمراً مريحاً وله العديد من التطبيقات الجديدة والرائعة للمستخدمين المتنقلين، إلا أنه للأسف يخلق أيضاً نقاط ضعف أمنية قد تشكل خطراً على الشبكة. بوضع مستقبِل سلبي على مقربة من مرسل لاسلكي، يمكن لذلك المستقبِل أن يحصل على نسخة من كل رزمة يتم إرسالها! ويمكن أن تتضمن تلك الرزم كل أنواع المعلومات الحسّاسة، بما في ذلك كلمات السر وأرقام الضمان الاجتماعي وأسراراً تجارية، ورسائل شخصية خاصّة. يُطلق علي أي مستقبِل سلبي يقوم بتسجيل نسخة من كل رزمة تمر من حوله "لاقط رزم".

كما تقدّم في الجزء 1-2، تقوم تقنيات الوصول السلكية أيضاً ببث الرزم، ومن ثم فهي عرضة للالتقاط أيضاً. لذا يمكن أن يعمل لاقط الرزم في البيئة السلكية أيضاً. ففي العديد من شبكات الإيثرنت المحلية بإمكان لاقط الرزم الحصول على نسخة من كل رزمة يتم إرسالها على الشبكة. علاوة على ذلك، بوسع شخص سيئ يتمكن من الوصول إلى موجّه أو وصلة الإنترنت الخاصة بمؤسسةٍ ما زرع برنامج لاقط رزم يقوم بعمل نسخة من كل رزمة بيانات تصل إلى المؤسسة أو تخرج منها. يتم بعد ذلك تحليل الرزم الملتقَطة بشكلٍ منفصل عن الإنترنت بحثاً عن المعلومات الحسّاسة.

تتوافر برامج التقاط الرزم مجاناً في مواقع الويب المختلفة وكمنتجات تجارية. بل لقد عُرف عن الأساتذة الذين يدرّسون مقررات عن الشبكات تخصيص تمارين مختبرات لكتابة برامج لالتقاط الرزم واستعادة البيانات في طبقة التطبيقات. في الواقع يستخدم مختبر Ethereal [Ethereal 2007] المصاحب لهذا الكتاب لاقط رزم من هذا النوع (انظر مختبر Ethereal التمهيدي في نهاية هذا الفصل).

ونظراً لأن برامج التقاط الرزم سلبية (أي أنها لا تحقن رزماً في قناة الاتصال) فإنه يصعب اكتشافها. لذلك فعندما نرسل الرزم على قناة لاسلكية، يجب أن نتقبل احتمال وجود شخص سيئ قد يسجّل نسخاً من رزمنا. ومن أفضل

طرق الحماية ضد التقاط الرزم (كما قد تكون توقعت) استخدام أسلوب التشفير، والذي سنتناوله مع علاقته بأمن الشبكات في الفصل الثامن.

انتحال شخصية آخرين ممن تأتمنهم

ستتكون لديك القناعة بعد قليل وأنت تشق طريقك عبر هذا الكتاب أنه من السهل جداً تكوين رزمة بقيم اختيارية لعنوان المصدر، ومحتوى الرزمة، وعنوان الوجهة، ثم إرسال تلك الرزمة المشكّلة يدوياً إلى الإنترنت، والتي ستقوم بواجبها خير قيام في توصيل الرزمة إلى وجهتها النهائية. تخيّل المستلم البريء (مثلاً موجّه على الإنترنت) الذي يتلقى مثل تلك الرزمة، ويأخذ عنوان المصدر الخطأ كما لو كان صحيحاً، وبعد ذلك ينفذ بعض الأوامر التي تضمّنتها محتويات الرزمة (كأن يعدّل في جداول التوجيه لديه). يطلق على القدرة على حقن الرزم إلى الإنترنت مع عنوان مصدر خاطئ تعبير تزييف العنوان (IP spoofing)، وهو مجرد واحد من عدة أساليب يمكن بواسطتها لمستخدم الإنترنت التنكر كمستخدمٍ آخر.

لحلّ هذه المشكلة سنحتاج لتطبيق أسلوب للتحقّق من هوية النقطة الطرفية، أي آلية تمكننا من التأكد بدقة مما إذا كانت الرسالة الواصلة لها نفس المنشأ الذي تدعيه. مرة أخرى نشجّعك على التفكير في كيفية تحقيق ذلك لتطبيقات وبروتوكولات الشبكة وأنت تتنقل بين فصول هذا الكتاب. سنستكشف الآليات المستخدمة للتحقّق من النقطة الطرفية في الفصل الثامن.

تعديل وحذف الرسائل

ننهي هذا الاستعراض القصير لهجمات الشبكة بوصف هجوم "رَجُل في الوسط" (man-in-the-middle attack). في هذا الصنف من الهجمات، يحشر الشخص السيئ نفسه على طريق الاتصال بين كيانين. دعنا نرمز لكياني الاتصال بأليس وبوب، والذي يمكن أن يكونا شخصين أو كيانين من كيانات الشبكة – كموجّهين أو خادمي بريد إلكتروني. يمكن أن يكون الشخص السيئ على سبيل المثال موجّهاً في طريق الاتصال، أو برنامجاً يستقر

على إحدى النهايتين الطرفيتين في طبقة منخفضة من رصة البروتوكولات. في هجوم "رجل في الوسط" يكون للشخص السيئ القدرة ليس فقط على التقاط كل الرزم التي تعبر بين بوب وأليس، ولكن أيضاً حقن وتعديل وحذف الرزم. في مفردات أمن الشبكات يمكن لهجوم الرجل في الوسط أن يؤثر على سلامة البيانات المرسلة بين أليس وبوب. كما سنرى في الفصل الثامن، فإن الآليات التي توفر السرية (الحماية ضد التقاط الرزم) وتوثيق النقطة الطرفية (التي تسمح للمستلم بالتحقق دون أدنى شك من هوية مرسل الرسالة) لا توفر بالضرورة سلامة البيانات. لذا فنحن بحاجة إلى مجموعة أخرى من التقنيات لتوفير سلامة البيانات.

في نهاية هذا الجزء، يجدر بنا أن نتأمل كيف أصبحت الإنترنت مكاناً غير آمن في المقام الأول. يكمن الجواب جوهرياً في أن الإنترنت صُمّمت أساساً لتكون كذلك، استناداً إلى نموذج "ربط مجموعة من المستخدمين تتوافر بينهم الثقة المتبادلة" [Blumenthal 2001]، وهو نموذج لا توجد فيه (من حيث التعريف) حاجة للأمن. إن العديد من سمات البنية الأصلية للإنترنت تعكس بجلاء هذا المفهوم للثقة المتبادلة. على سبيل المثال، فإن قدرة أي مستخدم على إرسال رزمة إلى أي مستخدم آخر هي القاعدة وليست الاستثناء الذي يُطلُب ويُمنَح. كما أن هوية المستخدم تُؤخذ على ظاهرها المُعلن، وليس على أساس أن القاعدة هي أن يتم التحقق منها.

غير أن إنترنت اليوم لا تقتصر بالتأكيد على "مستخدمين تتوافر بينهم الثقة المتبادلة". ومع ذلك فمستخدمو اليوم لايزالون بحاجة للاتصال رغم أنهم لا يأمن بعضهم بعضاً بالضرورة، وقد يرغبون في الاتصال بدون كشف هويتهم، وقد يتصلون بشكلٍ غير مباشر عبر طرف ثالث (مثلاً عن طريق ذاكرات الويب المخبأة (Web caches) والتي سندرسها في الفصل الثانى، أو وكلاء المساعدة لقابلية الحركة (Mobility-assisting agents) والتي سندرسها في الفصل السادس)، وقد يشكّون في العتاد أو البرمجيات أو حتى الهواء الذي يتصلون من خلاله. سنواجه العديد من التحديات المتعلقة بأمن

الشبكات كلما تقدمنا في هذا الكتاب، حيث يجب توفير دفاعات ضد التقاط الرزم، وضد انتحال شخصية نقطة طرفية، وضد هجوم رجل في الوسط، وضد هجوم حجب الخدمة الموزّع، وضد البرامج الخبيثة، وغيرها. وعلينا أن نتذكر أن الاتصال بين المستخدمين المؤتمنين بشكلٍ متبادل أصبح الآن الاستثناء وليس القاعدة، فمرحباً بك في عالم شبكات الحاسب العصرية!

7-1 تاريخ شبكات الحاسب والإنترنت

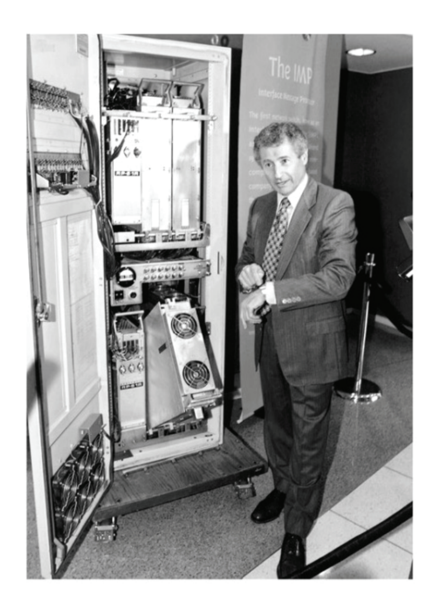
قدّمنا في الأجزاء من 1-1 إلى 1-6 استعراضاً لتقنية شبكات الحاسب والإنترنت. ويفترض أنك أصبحت الآن على إلمام كافٍ بالموضوع بشكلٍ عام. ومع ذلك، ولكي يكتمل لديك البعد التاريخي للصورة فإننا نستعرض هنا مقتطفات من التاريخ المثير لتطور الإنترنت [Segaller 1998].

1-7-1 تطور تحويل الرزم (الفترة 1961- 1972)

تعود بدايات تاريخ شبكات الحاسب وإنترنت اليوم إلى أوائل الستينيات من القرن الماضي، عندما كانت شبكة الهاتف هي شبكة الاتصالات العالمية المهيمنة. تذكّر أننا وضّحنا في الجزء 1-3 أن شبكة الهاتف تستخدم أسلوب تحويل الدوائر لنقل المعلومات من مرسل إلى مستقبل- وهو اختيار ملائم نظراً لأن الصوت يتم إرساله بمعدل ثابت بين المرسل والمستقبل. ونظراً للأهمية المتزايدة (والكلفة الباهظة) للحاسبات في أوائل الستينيات وظهور تقنية استخدامها بالمشاركة في الوقت، فقد كان من الطبيعي (على الأقل باستدراك متأخر مثالي!) التفكير في طريقة لتوصيل الحاسبات ببعضها لكي يتسنى الاشتراك في استعمالها بين عدة مستخدمين موزّعين جغرافياً. إن حركة المرور المتولدة من قِبَل مثل هؤلاء المستخدمين يغلب عليها أن تكون على فترات متقطعة من النشاط، مثلاً إرسال أمر إلى حاسب بعيد، تتبعه فترات خمول بينما ينتظر المستخدم إجابة أو يتأمّل الردّ الذي تسلّمه من الحاسب.

شرَعت ثلاث مجموعات بحث مستقلة في أنحاء مختلفة حول العالم، كل منها بمعزل عن الآخرين [Leiner 1998]، في تطوير تقنية تحويل رزم البيانات كبديل كفء وقوي لتحويل الدوائر. نُشِر أول بحث عن تقنيات تحويل رزم البيانات ليونارد كلينروك [Kleinrock 1961; Kleinrock 1964]، عندما كان طالب دراسات عليا في MIT. باستخدام نظرية طوابير الانتظار، أثبتت أبحاث كلينروك بشكلٍ رائع فعالية أسلوب تحويل رزم البيانات في حالة مصادر حركة المرور المتقطعة. في عام 1964 بدأ بول باران [Baran 1964] في معهد راند دراسة استعمال تحويل رزم البيانات للنقل الآمن للصوت على الشبكات العسكرية، وفي مختبر الفيزياء الوطني (NPL) في إنجلترا كان دونالد ديفيز وروجر سكانتلبري يطوّران أفكارهم أيضاً بخصوص تحويل رزم البيانات.

لقد وضعت تلك الأبحاث الثلاثة في MIT وRand وNPL حجر الأساس لإنترنت اليوم، ولكن للإنترنت أيضاً تاريخ طويل مع أسلوب "دعنا نبنيها ونثبت أنها تعمل" والذي يعود إلى الستينيات أيضاً. اضطلع جوزيف لكلايدر [DEC 1990] ولورانس روبرتس، وكلاهما زملاء لكلينروك في MIT، بقيادة برنامج لعلوم الحاسبات في وكالة مشاريع البحوث المتقدمة (ARPA) في الولايات المتحدة، كما نشر روبرتس خطة شاملة لشبكة أربانت (ARPAnet) [Roberts 1967]، وهي أول شبكة حاسب بتحويل الرزم وتعتبر سلفاً مباشراً لإنترنت اليوم. كانت محوّلات الرزم المبكرة تعرف بمعالجات رسائل الواجهات (IMPs)، وقد مُنح عقد تصنيعها لشركة BBN يوم عيد العمال في 1969. تم تركيب أول محوّل في جامعة كاليفورنيا في لوس أنجلوس (UCLA) تحت إشراف كلينروك، وبعد ذلك بقليل تم تركيب ثلاثة محوّلات إضافية في معهد ستانفورد للأبحاث، وجامعة كاليفورنيا في سانتا باربرة، وجامعة يوتا (الشكل 1-22). تضمنت الشبكة الجديدة التي شكّلت بداية الإنترنت أربع عقد بنهاية عام 1969. يتذكّر كلينروك أول استعمال للشبكة لتحقيق اتصال عن بعد من UCLA إلى معهد ستانفورد للأبحاث، حين أدى ذلك إلى تعطل النظام [Kleinrock 2004].



الشكل 1-22 باحث الشبكات ليونارد كلينروك يقف أمام نموذج مبكر من معالجات رسائل الواجهات (IMPs) [Mark J. Terril, AP/Wide World Photos].

بحلول عام 1972 نمت شبكة أربانت لتضم حوالى 15 عقدة، وتم إجراء أول استعراض عملي عليها للجمهور بواسطة روبرت كاهن في مؤتمر 1972 الدولي عن اتصالات الحاسب. تم إنجاز أول بروتوكول مضيف إلى مضيف (طرف إلى طرف) للاستخدام بين أنظمة أربانت الطرفية، والذي عرف باسم بروتوكول التحكم في الشبكة (NCP) [RFC 001]. وبوجود بروتوكول طرف إلى طرف، أصبح من الممكن كتابة برامج التطبيقات. قام راي توملينسون في شركة BBN بكتابة أول برنامج للبريد الإلكتروني في عام 1972.

1-7-2 الشبكات ذات الملكية الخاصة، والتوصيل ما بين الشبكات (الفترة 1972 – 1980)

كانت شبكة أربانت شبكة واحدة ومغلقة. ولكي تستطيع الاتصال بأحد مضيفات الشبكة كان لزاماً عليك أن تكون موصلاً بالفعل بإحدى معالجات رسائل الواجهة (IMP) على الشبكة. في أوائل السبعينيات وحتى

منتصفها بدأت شبكات جديدة ومستقلة لتحويل الرزم تظهر إلى حيز الوجود جنباً إلى جنب مع شبكة أربانت:

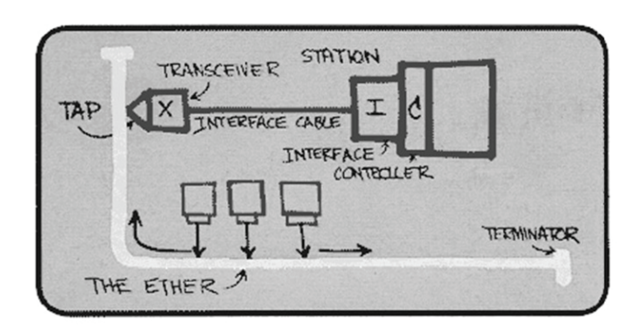
- شبكة ألوهانت ALOHAnet: شبكة مايكروويف تربط الجامعات على جزر ولاية هاواي [Abramson 1970]، بالإضافة إلى شبكات الرزم للأقمار الصناعية (packet satellite) [RFC 829] وشبكات رزم الراديو (packet radio networks) التابعة لوكالة مشاريع البحوث المتطورة للدفاع (DARPA) [Kahn 1978].
- شبكة تيلنت Telenet: شبكة BBN تجارية لتحويل الرزم مبنية على أساس تقنية أربانت.
- شبكة سيكليدز Cyclades: شبكة فرنسية لتحويل الرزم من تطوير لويس بوزين [Think 2007].
- شبكات المشاركة في الوقت: مثل Tymnet وشبكة GE لخدمات المعلومات وغيرها في أواخر الستينيات وأوائل السبعينيات [Schwartz 1977].
- شبكة SNA من IBM (1969-1974): وقد تزامنت بالتوازي مع أعمال تطوير شبكة أربانت [Schwartz 1977].

ومع تزايد عدد الشبكات يمكن أن ندرك بأن الوقت كان قد حان لتطوير بنية عامة لتوصيل الشبكات ببعضها. قام كل من فينتون سيرف وروبرت كاهن [Cerf 1974] بالعمل الرائد لربط الشبكات (بدعم مالي من وكالة DARPA، والذي كان يهدف أساساً لتكوين شبكة من الشبكات، وتم صياغة المصطلح الجديد internetting لوصف هذا العمل.

لقد تم تجسيد تلك المبادئ المعمارية الجديدة في بروتوكول التحكم في الإرسال (TCP). ومع ذلك فقد كانت الإصدارات المبكّرة من بروتوكول TCP مختلفة كثيراً عن TCP الذي نعرفه اليوم. جمعت تلك الإصدارات الأولى بين نظام موثوق لنقل البيانات بالترتيب عن طريق إعادة الإرسال بين الأنظمة

الطرفية (والذي لا يزال يمثل جزءاً من TCP اليوم) ووظائف التوجيه (والتي تؤدى اليوم من خلال بروتوكول IP). أدت التجارب الأولى على بروتوكول TCP والقناعة بأهمية وجود خدمة نقل من طرف إلى طرف غير موثوقة وبدون تحكم في التدفق لاستخدامه في التطبيقات من نوع إرسال الصوت على شكل رزم، إلى فصل بروتوكول IP عن بروتوكول TCP وتطوير بروتوكول جديد هو UDP. وبذلك كانت بروتوكولات الإنترنت الرئيسية الثلاثة التي نعرفها اليوم (TCP وUDP وIP) في مكانها من حيث المفهوم بحلول نهاية السبعينيات.

بالإضافة إلى أبحاث DARPA المتعلقة بالإنترنت، شهدت تلك الفترة العديد من النشاطات المهمة الأخرى لربط الشبكات. ففي ولاية هاواي كان نورمان أبرامسون يطوّر شبكة الراديو ALOHAnet المعتمدة على نقل الرزم للسماح لعدة مواقع متباعدة على جزر هاواي من الاتصال مع بعضها البعض. كان بروتوكول ALOHA [Abramson 1970] أول بروتوكول للوصول المتعدّد يسمح للمستخدمين الموزّعين جغرافياً بالاشتراك في وسط اتصالات إذاعي واحد (عبر ترددات الراديو). بعد ذلك اعتمد ميتكالف وبوجز على بروتوكول أبرامسون للوصول المتعدد في تطوير نظام الإيثرنت [Metcalfe 1976] لشبكات إذاعة مشتركة أساسها السلك (انظر الشكل 1-23). من المثير للانتباه أن الدافع لتطوير بروتوكول الإيثرنت بواسطة ميتكالف وبوجز كان الحاجة لتوصيل عدة حاسبات شخصية وطابعات وأقراص تخزين مشتركة [Perkins 1994]. أي أنه منذ ثلاثين سنة مضت ـ أي قبل ثورة الحاسب الشخصي وانفجار الشبكات ـ كان ميتكالف وبوجز يضعان الأساس لشبكات الحاسب الشخصي المحلية المعروفة اليوم. لعبت تقنية الإيثرنت دوراً مهماً كذلك في تربيط الشبكات (Internetworking). وبزيادة عدد تلك الشبكات المحلية وانتشارها أصبحت الحاجة ملحّة لتربيط تلك الشبكات مع بعضها البعض بصورة متزايدة. سنتناول الإيثرنت وALOHA وغيرها من تقنيات الشبكات المحلية الأخرى بالتفصيل في الفصل الخامس.



الشكل 1-23 تصوّر "ميتكالف" المبدئي للإيثرنت.

1-7-3 انتشار الشبكات (الفترة 1980-1990)

بحلول نهاية السبعينيات كان هناك حوالي مائتي مضيف موصلين بشبكة ARPAnet. وفي نهاية الثمانينيات قفز عدد المضيفات الموصلة بالإنترنت العامة (وهي اتحاد من الشبكات يشبه إلى حدٍّ كبير إنترنت اليوم) ليصل إلى مائة ألف. ومن ثم شهدت فترة الثمانينيات نمواً هائلاً.

نتج الجزء الأكبر من ذلك النمو من عدة جهود محددة لإنشاء شبكات للحاسب تربط الجامعات مع بعضها. كما وفرت شبكة BITNET خدمات البريد الإلكتروني ونقل الملفات بين عدة جامعات في شمال شرق الولايات المتحدة، كما تم إنشاء شبكة CSNET (شبكة علوم الحاسبات) لربط باحثي الجامعات الذين ليس لديهم طريقة للوصول لشبكة أربانت. وفي عام 1986 تأسست شبكة مؤسسة العلوم القومية NSFNET لتوفير وصول إلى مراكز الحاسبات الكبرى المدعومة من قبل المؤسسة. بدأت الشبكة بسرعة عمود فقري مقدارها 56 كيلوبت/ثانية وارتفعت لتبلغ 1.5 ميجابت/ثانية في نهاية العقد حيث أصبحت الشبكة بمثابة عمود فقري أساسي لربط الشبكات الإقليمية.

في مجموعة أربانت كانت العديد من القطع النهائية تأخذ مكانها لتشكل بنية الإنترنت كما نعرفها اليوم. شهد الأول من يناير(كانون الثّاني) عام 1983 التدشين الرسمي لطقم البروتوكول TCP/IP كبروتوكول المضيف المعياري الجديد لشبكة ARPAnet (ليحل محل بروتوكول NCP). كان يوم الانتقال [RFC 801] من NCP إلى TCP/IP حدثاً هاماً، حيث تعين على المضيفات على الشبكة التحول إلى TCP/IP ابتداء من ذلك اليوم. في أواخر الثمانينيات، تم إدخال إضافات مهمة على TCP لتوفير تحكم في الازدحام أساسه المضيف [Jacobson 1988]، كما تم أيضاً تطوير بروتوكول DNS [RFC 1034] للتحويل مابين أسماء مواقع الإنترنت (بصيغتها السهلة للتعامل البشري مثل gaia.cs.urnass.edu) والعنوان الرقمي المناظر بطول 32 بت على الشبكة.

بالتوازي مع أعمال تطوير أربانت (والتي كانت في الغالب جهوداً أمريكيةً)، أطلق الفرنسيون في أوائل الثمانينيات مشروع Minitel، والذي كان بمثابة خطة طموحة لجلب شبكة البيانات إلى كل منزل. شمل نظام Minitel المدعوم من قِبَل الحكومة الفرنسية شبكة عامة لتحويل الرزم (مستندة على طقم البروتوكولات X.25)، وخادمات Minitel، ومحطات طرفية رخيصة بمودمات مدمجة ذات سرعة منخفضة. حقق مشروع Minitel نجاحاً عظيماً في عام 1984 عندما منحت الحكومة الفرنسية محطة طرفية Minitel مجاناً لكل عائلة فرنسية ترغب في ذلك. تضمنت شبكة Minitel مواقع مجانية كموقع دليل الهاتف، بالإضافة إلى مواقع خاصة توفر خدماتها نظير رسوم تعتمد على حجم التعامل. وفي أوج انتشاره في منتصف التسعينيات كان النظام يوفر أكثر من 20 ألف خدمة تتراوح من التعاملات البنكية من المنازل إلى قواعد بيانات بحث متخصصة، وكان النظام يُستخدم من قِبَل 20٪ من سكان فرنسا، حيث ولّد دخلاً تجاوز بليون دولار أمريكي كل سنة، ووفر 10 آلاف وظيفة. وجد نظام Minitel طريقه إلى نسبة كبيرة من المنازل الفرنسية لفترة عشرة سنوات قبل أن يسمع أكثر الأمريكان عن الإنترنت.

1-7-4 انفجار الإنترنت (فترة التسعينيات)

شهدت التسعينيات من القرن الماضي عدة أحداث جسدت التطور المستمر للإنترنت وبشرت بقرب الاستخدام التجاري لها. لقد اختفت شبكة ARPAnet (سلف الإنترنت) من الوجود، ونمت الشبكة العسكرية MILNET وشبكة البيانات الدفاعية في الثمانينيات لتحمل أغلب حركة مرور وزارة الدفاع الأمريكية، وبدأت شبكة المؤسسة القومية للعلوم NSFNET في العمل كعمود فقري يوصل شبكات إقليمية في الولايات المتحدة وشبكات وطنية في الخارج. في عام 1991 رفعت شبكة NSFNET قيودها على استخدام الشبكة للأغراض التجارية. وفي عام 1995 تم إحالة شبكة NSFNET نفسها إلى التقاعد حيث تولى موفرو خدمة الإنترنت التجاريين نقل حركة المرور على العمود الفقري للإنترنت.

ومع ذلك فقد كان الحدث الرئيسي في التسعينيات ظهور تطبيق شبكة الويب العالمية ((World Wide Web (WWW)) الذي جلب الإنترنت إلى المنازل والأعمال التجارية للملايين من البشر في مختلف أنحاء العالم. عمل الويب كمنصة لتمكين ونشر المئات من التطبيقات الجديدة التي نعتبرها الآن شيئاً مفروغاً منه. للاطلاع على تاريخ مختصر للأيام الأولى للويب راجع [W3C 1995].

تم اختراع الويب في معامل المنظمة الأوروبية للبحوث النووية (CERN) بسويسرا من قِبَل تيم بيرنرز - لي [Berners-Lee 1989] في الفترة ما بين 1989 و1991، مستنداً على أفكار فانيفار بوش عن النص التشعبي (hypertext) في الأربعينيات [Bush 1945] وتيد نيلسن في الستينيات [Xanadu 2007]. طوّر تيم بيرنرز- لي وشركاه الإصدارات الأولية من لغة HTML وبروتوكول HTTP وخادم ويب وبرنامج تصفّح - أي المكونات الأربعة الرئيسية للويب. باقتراب نهاية 1992 كان هناك حوالي مائتي خادم ويب موصل بالإنترنت، ومع ذلك فإن هذه المجموعة من الخادمات لم تكن سوى بشائر لما كانت تحمله الأيام

القادمة. في غضون ذلك الوقت كان هناك عدد من الباحثين يطوّرون متصفحات ويب بواجهات مستخدم رسومية (GUI)، من بينهم مارك أندرسين الذي قاد عملية تطوير المتصفح الشهير موزايك (Mosaic). وفي 1994 أسس مارك أندرسين وجيم كلارك شركة Mosaic Communications والتي أصبحت فيما بعد Netscape Communications Coportation [Cusumano 1998; Quittner 1998]. بحلول عام 1995 كان طلاب الجامعات يستخدمون متصفّحات نيتسكيب وموزايك لتصفّح الويب بشكلٍ يومي. وقريباً من ذلك الوقت كانت الشركات الصغيرة والكبيرة قد بدأت في تشغيل خادمات الويب والقيام بأعمال تجارية على الويب. ففي عام 1996، بدأت شركة مايكروسوفت في إنتاج المتصفّحات، مما أشعل حرب المتصفّحات بين نيتسكيب ومايكروسوفت، والتي ربحتها مايكروسوفت بعد سنوات قليلة [Cusumano 1998].

كان النصف الثاني من التسعينيات فترة نمو وإبداع هائلين للإنترنت، حيث قامت الشركات الكبيرة وآلاف من الشركات البادئة بتطوير العديد من منتجات وخدمات الإنترنت. تواصل تطور تطبيقات بريد الإنترنت الإلكتروني، حيث ظهرت برامج لقراءة البريد غنية بالمزايا كدفاتر العناوين والتعامل مع الملحقات والروابط التشعبية ونقل مواد الوسائط المتعددة. وبحلول نهاية الألفية الثانية كانت الإنترنت تدعم المئات من التطبيقات المشهورة، بما في ذلك التطبيقات الأربعة التالية التي اكتسحت الساحة:

- البريد الإلكتروني، بما في ذلك الملحقات وبريد الويب الإلكتروني لتسهيل الوصول.
- الويب، بما في ذلك تصفّح الويب وتجارة الإنترنت.
- الرسائل الفورية بقوائم الاتصال من ابتكار شركة ICQ
- مشاركة النظائر لملفات MP3 من ابتكار شركة Napster.

ومما يثير الانتباه أن التطبيقين الأولين جاءا من العاملين في مجال الأبحاث، بينما قام بتطوير الأخيرين بضعة رجال أعمال شباب.

كانت الفترة من عام 1995 إلى عام 2001 فترة تقلبات سريعة للإنترنت في الأسواق المالية. حتى قبل أن تحقق أرباحاً، طرحت مئات شركات الإنترنت البادئة أسهمها لأول مرة للجمهور وبدأ تداولها بسوق الأسهم المالية. قُيِّمت العديد من الشركات ببلايين الدولارات بدون امتلاك أي موارد ذات قيمة للعائدات. انهارت أسهم الإنترنت المالية في العامين 2000 و2001 وأُغلقت العديد من الشركات البادئة. وعلى الرغم من ذلك فقد خرج عدد من الشركات كفائزين كبار في مجال الإنترنت، من بينها مايكروسوفت وسيسكو وياهوو وإي- باي، وجوجل، وأمازون.

5-7-1 التطورات الحديثة

يتواصل الإبداع في شبكات الحاسب بخطى سريعة. حيث تشهد الفترة الأخيرة تقدماً على كل الجبهات، بما في ذلك انتشار التطبيقات الجديدة وتوزيع المحتوى وهاتف الإنترنت والإرسال بسرعات أعلى في الشبكات المحلية وتطوير موجّهات أسرع. لكن هناك ثلاثة مجالات للتطور تستحق انتباها خاصاً: انتشار شبكات الوصول السريع (بما في ذلك الوصول اللاسلكي) وأمن الشبكات وشبكات النظائر.

كما ذكرنا في الجزء 1-2 مهّد الطلب المتزايد للوصول السكني بحيز ترددي عريض بالإنترنت عن طريق مودم الكبل وDSL لمجموعة ثرية من التطبيقات الجديدة للوسائط المتعددة، بما في ذلك: الصوت والفيديو على IP [Skype 2007] والاشتراك في أفلام الفيديو [YouTube 2007] والتلفزيون على IP [PPLive 2007]. إن الانتشار المتزايد لشبكات WiFi العامة السريعة (11 ميجابت/ثانية أو أعلى) والوصول للإنترنت عن طريق شبكات الهاتف الخلوي بسرعة متوسطة (مئات الكيلوبتات/ثانية) لا يسهّل فقط بقاء المستخدم موصلاً بالإنترنت بشكلٍ دائم، ولكن أيضاً يمكّن لمجموعة جديدة ومثيرة من الخدمات المتعلقة بمواقع معينة. سنغطي الشبكات اللاسلكية وقابلية الحركة في الفصل السادس.

في أعقاب سلسلة من هجمات حجب الخدمة على خادمات الويب البارزة في أواخر التسعينيات وانتشار هجمات الدودة (كدودة Blaster)، أصبح أمن الشبكات موضوعاً في غاية الأهمية. أدّت تلك الهجمات إلى تطوير أنظمة لاكتشاف محاولات الاختراق، وتوفير إنذار مبكّر قبل الهجوم وبرامج حماية (firewalls) تقوم بترشيح حركة المرور واستبعاد غير المرغوب منها قبل أن يدخل الشبكة. سنغطّي عدداً من المواضيع الهامة المتعلقة بأمن الشبكات في الفصل الثامن.

أما الإبداع الأخير الذي نوليه عناية خاصة هنا فهو شبكات النظائر (P2P). تستغل شبكات النظائر الموارد المتوفرة في حاسبات المستخدمين كوحدات التخزين، والمحتوى، وخطوات التنفيذ بوحدات المعالجة المركزية، والحضور البشري، لتحقيق استقلال ذاتي مهم عن الخادمات المركزية على الشبكة. عادةً ما يكون توصيل المستخدمين (وبمعنى آخر النظائر) بشكلٍ متقطّع. كان هناك العديد من قصص نجاح شبكات النظائر في السنوات القليلة الماضية، بما في ذلك مشاركة الملفات (Napster وKazaa وGnutella وeDonkey وLimeWire، إلخ)، وتوزيع الملفات (BitTorrent)، وهاتف الإنترنت (Skype)، وتلفزيون الإنترنت (PPLive وppStream). سنتناول العديد من تطبيقات شبكات النظائر هذه في الفصل الثاني.

8-1 الخلاصة

تناولنا في هذا الفصل كمية هائلة من المادّة العلمية! استعرضنا المكونات المختلفة من العتاد والبرامج التي تكوّن الإنترنت بشكلٍ خاص وشبكات الحاسب عموماً. بدأنا على حافة الشبكة، حيث ألقينا نظرةً على الأنظمة والتطبيقات الطرفية وعلى خدمات نقل البيانات التي توفّرها الشبكة للتطبيقات التي تشغّلها تلك الأنظمة الطرفية. تناولنا أيضاً تقنيات طبقة ربط البيانات وأوساط التوصيل المادية التي تستخدم عادةً في شبكة الوصول. بعد ذلك دلفنا بعمق أكثر داخل الشبكة باتجاه القلب، حيث تعرفنا على تحويل رزم البيانات وتحويل الدوائر كأسلوبين أساسيين لنقل البيانات خلال شبكة اتصال،

واستعرضنا نقاط القوة والضعف لكلٍّ منهما. تناولنا بعد ذلك تركيب الإنترنت العالمية حيث عرفنا أن الإنترنت هي شبكة من الشبكات، ورأينا أن هيكل الإنترنت الهرمي الذي يضم موفري خدمة الإنترنت من الطبقات العليا والدنيا قد مكّنها من التوسع لتضم آلاف الشبكات.

في الجزء الثاني من هذا الفصل التمهيدي، تناولنا عدة مواضيع أساسية في مجال شبكات الحاسب. استعرضنا أولاً أسباب التأخير، ثم فقد الرزم، والطاقة الإنتاجية في شبكة تحويل الرزم. طوّرنا بعد ذلك نماذج كمية بسيطة للتأخير المتعلق بالإرسال، والانتقال، وصفوف الانتظار، وكذلك الطاقة الإنتاجية، وسنستخدم نماذج التأخير في صفوف الانتظار بشكلٍ مكثف في تمارين الواجب المنزلي في كافة أجزاء هذا الكتاب. بعد ذلك درسنا طبقات البروتوكولات ونماذج الخدمة المرتبطة بها، وشرحنا المبادئ الأساسية للبنية المعمارية للشبكات، والتي سنشير إليها بعد ذلك في مختلف أنحاء الكتاب. استعرضنا أيضاً بعض أنواع الهجمات الأمنية الأكثر انتشاراً في إنترنت اليوم. وأخيراً ختمنا مقدمتنا عن الشبكات بتاريخ مختصر لشبكات الحاسب.

يعتبر الفصل الأول في حدِّ ذاته مقرراً مصغراً في شبكات الحاسب، فقد غطينا كماً كبيراً من المواضيع الأساسية في هذا الفصل! وإذا كنت تشعر بشيء من الارتباك إزاء هذا الكم الهائل من المعلومات فلا تقلق!، ففي الفصول التالية سنزور كل هذه الأفكار من جديد لنغطيها بتفصيل أكثر. الآن نتمنى فقط أن تغادر هذا الفصل بحسٍ متنامٍ للأجزاء التي تكوّن شبكة، وبرغبة مستمرة في تحسين استيعابك لمفردات الشبكات، وبهمة متزايدة دوماً للتعلم أكثر عن الشبكات. تلك هي المهمة الملقاة على عاتقنا في الجزء الباقي من هذا الكتاب، ولكن لا تتردد في الرجوع إلى هذا الفصل في أي وقت.

خارطة محتويات هذا الكتاب

قبل بدء أي رحلة عليك دائماً أن تلقي نظرة على الخارطة لكي تتعرف على الطرق والتقاطعات الرئيسة التي ستتعامل معها خلال الرحلة. في رحلتنا التي نحن

بصدد بدئها مع هذا الكتاب، وجهتنا النهائية هي فهم عميق لـ: ماذا وكيف ولماذا شبكات الحاسب. خارطتنا في هذه الرحلة هي سلسلة فصول هذا الكتاب:

1- شبكات الحاسب والإنترنت
2- طبقة التطبيقات
3- طبقة النقل
4- طبقة الشبكة
5- طبقة ربط البيانات والشبكات المحلية
6- الاتصال اللاسلكي وقابلية الحركة
7- شبكات الوسائط المتعددة
8- أمن شبكات الحاسب
9- إدارة الشبكة

تعتبر الفصول من الثانى إلى الخامس الفصول الأربعة الرئيسة في هذا الكتاب. ستلاحظ أن هذه الفصول مهيكلة حول الطبقات الأربعة العليا من رصة بروتوكولات الإنترنت ذي الطبقات الخمس، أي بمعدل فصل واحد عن كل طبقة. لاحظ أيضاً أن رحلتنا ستبدأ في قمة رصة بروتوكولات الإنترنت أي طبقة التطبيقات، ثم نتجه لأسفل خلال رصة البروتوكولات. السبب الجوهري وراء قطع الرحلة من أعلى إلى أسفل وليس العكس هو أنه عندما نفهم التطبيقات سيكون بوسعنا فهم خدمات الشبكة اللازمة لدعم تلك التطبيقات. بعد ذلك سيمكننا بالتالي فحص الطرق المختلفة التي يمكن بها إنجاز مثل تلك الخدمات من قِبَل البنية المعمارية للشبكة، وبهذا تكون تغطية التطبيقات في البداية حافزاً لمتابعة بقية فصول الكتاب.

يركّز النصف الثاني من فصول الكتاب، أي من الفصل السادس إلى الفصل التاسع على أربعة مواضيع مهمة جداً (ومستقلة إلى حدٍّ ما) في شبكات الحاسب الحديثة. ففي الفصل السادس نتناول الشبكات اللاسلكية وقابلية

الحركة، بما ذلك الشبكات المحلية اللاسلكية (ومنها WiFi، وWiMAX، وBluetooth)، وشبكات الهاتف الخلوي (ومنها GSM)، وقابلية الحركة في كلٍّ من شبكات IP وشبكات GSM. وفي الفصل السابع (شبكات الوسائط المتعددة) نتناول تطبيقات الصوت والفيديو، كهاتف الإنترنت، ومؤتمرات الفيديو، وتشغيل مواد الوسائط المتعددة المخزّنة. سندرس أيضاً كيف يمكن تصميم شبكة لتحويل الرزم لتقديم مستوى متسق من جودة الخدمة لتطبيقات الصوت والفيديو. وفي الفصل الثامن (أمن الشبكات) نتناول أساسيات التشفير وأمن الشبكات لكي نرى كيف تُطبّق النظرية الأساسية في تشكيلة واسعة من سياقات الإنترنت. أما الفصل الأخير (إدارة الشبكات) فيتناول القضايا الرئيسية في إدارة الشبكات بالإضافة إلى بروتوكولات الإنترنت الأساسية المستخدمة لذلك.

أسئلة وتمارين وتدريبات الفصل الأول

❖ أسـئلة مراجعة

• الجزء 1-1

1. ما الفرق بين المضيف والنظام الطرفي؟ اذكر الأنواع المختلفة من الأنظمة الطرفية. هل خادم الويب نظام طرفي؟
2. تُستخدم كلمة "بروتوكول" كثيراً لوصف العلاقات الدبلوماسية. اعط مثالاً لبروتوكول دبلوماسي.

• الجزء 1-2

3. ما هو برنامج الزبون؟ ما هو برنامج الخادم؟ هل يقوم برنامج الخادم بطلب الخدمات من برنامج الزبون وإرسالها إليه؟
4. اذكر ستة أنواع من تقنيات الوصول. صنِّف كل نوعٍ منها كوصول سكني، وصول من الشركات/المؤسسات، أو وصول نقّال.
5. هل معدل الإرسال في شبكات الوصول الهجينة HFC مخصص لكل مستخدم على حدة، أم مشترك بين المستخدمين جميعاً؟ هل من الممكن حدوث الاصطدامات على وصلة HFC النازلة من الإنترنت؟ علل إجابتك.
6. اذكر تقنيات الوصول المتاحة في مدينتك. لكل نوع من تقنيات الوصول، اذكر المعدلات المعلنة في الاتجاه النازل من الإنترنت والصاعد إليها، وكذلك الرسوم الشهرية للخدمة.
7. ما هو معدل الإرسال على شبكة الإيثرنت المحلية؟ لمعدل إرسال معيّن، هل بوسع كل مستخدم على الشبكة الإرسال بشكلٍ مستمر بهذا المعدل؟
8. اذكر بعض الوسائط المادية التي يمكن أن تستخدمها الإيثرنت.
9. تستخدم شبكات الوصول السكني المودمات الهاتفية، وشبكات HFC الهجينة، وخطوط DSL. لكل واحدة من تقنيات الوصول تلك، اذكر المدى المتاح لمعدلات التردد، ووضح ما إذا كان معدل الإرسال مشترك بين المستخدمين أم مخصص لكل مستخدم على حدة.
10. صف أكثر تقنيات الوصول للإنترنت شيوعاً هذه الأيام، ثم قارن بين تلك التقنيات.

• الجزء 1-4

11. ما الميزة التي تتمتع بها شبكة تحويل الدوائر مقارنةً بشبكة تحويل الرزم؟ في شبكة بتحويل الدوائر، ما ميزة الإرسال المتعدد بتقسيم الوقت (TDM) على الإرسال المتعدد بتقسيم التردد (FDM)؟
12. لماذا يُقال: إنّ تحويل الرزم يتضمن إرسالاً متعدداً إحصائياً؟ قارن بين الإرسال المتعدد الإحصائي والإرسال المتعدد بتقسيم التردد (FDM).
13. افترض أن هناك محوّل رزم واحد بين مضيف مرسل ومضيف مستقبل. معدل الإرسال بين المضيف المرسل والمحوّل هو R_1 وبين المحوّل والمضيف المستقبل هو R_2. افترض أن المحوّل يستخدم تحويل الرزم بأسلوب "تخزين ثم إرسال"، ما هو التأخير الكلي من طرف إلى طرف لإرسال رزمة طولها L بتاً (اهمل تأخير الانتظار في الصف، وتأخير الانتقال، وتأخير المعالجة).
14. ما الفرق الأساسي الذي يميِّز بين موفر خدمة إنترنت من الطبقة-1 وآخر من الطبقة -2؟
15. افترض أن عدة مستخدمين يشتركون في وصلة لها معدل إرسال قدره 2 ميجابت/ثانية. افترض أيضاً أن كل مستخدم يرسل فقط أثناء 20٪ من الوقت. (انظر المناقشة حول الإرسال المتعدد الإحصائي في الجزء 1-3).
    a. كم عدد المستخدمين الذين يمكن استيعابهم باستخدام تحويل الدوائر؟
    b. لبقية السؤال، افترض استخدام تحويل الرزم. لماذا لن يكون هناك تأخير انتظار في الصف قبل الوصلة إذا قام اثنان أو أقل من المستخدمين بالإرسال في نفس الوقت؟ لماذا سيكون هناك تأخير انتظار في الصف إذا قام ثلاثة مستخدمين بالإرسال في نفس الوقت؟
    c. احسب احتمال قيام مستخدم بعينه بالإرسال.
    d. افترض الآن أن هناك ثلاثة مستخدمين. احسب احتمال قيام المستخدمين الثلاثة في وقت معين بالإرسال معاً. احسب الجزء من الوقت الذي تأخذ فيه أطوال صفوف الانتظار في الازدياد.

• الجزء 1-3

16. خذ في الاعتبار رزمة تنتقل من مصدر مضيف إلى مصدر وجهة عبر مسار ثابت. اذكر مكوّنات التأخير من طرف إلى طرف. أي تلك المكونات ثابت وأيها متغير؟

17. قم بتجريب برنامج جافا التفاعلي الخاص بالمقارنة بين تأخير الإرسال وتأخير الانتقال (Transmission Versus Propagation Delay) على موقع الويب المصاحب لهذا الكتاب. من بين معدلات الإرسال، وتأخير الانتقال، وأطوال الرموز، أوجد التبديلة التي تضمن أن ينتهي المرسِل من الإرسال قبل أن يصل أول بت من الرزمة إلى المستقبِل. أوجد تبديلة أخرى تصل فيها أول بت من الرزمة إلى المستقبِل قبل أن ينتهي المرسل من الإرسال.
18. كم يستغرق انتقال رزمة طولها 1000 بايت عبر وصلة طولها 2500 كم، بمعدل إرسال 2 ميجابت/ثانية، وسرعة انتقال 2.5×10^8 متر/ثانية؟ بشكلٍ أكثر عموماً، كم يستغرق انتقال رزمة طولها L عبر وصلة طولها d، بمعدل إرسال R، وسرعة انتقال s متر/ثانية؟ هل يعتمد هذا التأخير على طول الرزمة؟ هل يعتمد هذا التأخير على معدل الإرسال؟
19. افترض أن مضيف A يريد إرسال ملف كبير إلى مضيف B. يتكون المسار من A إلى B من ثلاث وصلات لها معدلات الإرسال: R_1 = 500 كيلوبت/ثانية، R_2 = 2 ميجابت/ثانية، R_3 = 1 ميجابت/ثانية.
    a. بافتراض عدم وجود أي حركة مرور بيانات أخرى في الشبكة، ما هي الطاقة الانتاجية لنقل الملف؟
    b. افترض أن الملف يتكون من 4 مليون بايت. كم من الوقت سيستغرق إرسال الملف إلى مضيف B؟
    c. كرر (a) و (b) أعلاه، ولكن مع تقليل R_2 إلى 100 كيلوبت/ثانية.
20. افترض أن نظاماً طرفياً A يريد إرسال ملف كبير إلى نظام طرفي B. على مستوى عالٍ جداً اشرح كيف يقوم النظام A بتكوين رزم من الملف. عندما تصل رزمة من الملف إلى محوّل رزم، أي المعلومات الموجودة في الرزمة سيستخدمها المحوّل لتحديد الوصلة الخارجة التي سيمرر الرزمة إليها؟ لماذا يشبه تحويل الرزم في الإنترنت قيادة سيارة من مدينة إلى أخرى مع الاسترشاد بالاتجاهات على الطريق؟
21. قم بتجريب برنامج جافا الخاص بالانتظار في الصفوف والفقد (Queuing and Loss) على موقع الويب المصاحب لهذا الكتاب. ما هو أقصى معدل للبث (emission rate) وأقل معدل للإرسال (transmission rate)؟ عند هذين المعدلين، ماهي كثافة حركة مرور البيانات؟ قم بتشغيل البرنامج بتلك المعدلات وحدد الوقت اللازم لكي يبدأ الفقد في الرزم. كرر التجربة مرة أخرى وحدّد من جديد الوقت اللازم لكي يبدأ الفقد في الرزم. هل النتائج مختلفة؟ علل إجابتك.

- **الجزء 1-5**

22. اذكر خمس مهام يمكن أن تقوم بها طبقة من طبقات البروتوكول. هل يمكن أن تقوم طبقتان (أو أكثر) بتنفيذ مهمة (أو أكثر) من تلك المهام؟
23. ما هي الطبقات الخمس في رصة بروتوكول الإنترنت؟ ماهي المسئوليات الرئيسة التي تضطلع بها كل طبقة؟
24. ما هي: رسالة طبقة التطبيقات؟ قطعة طبقة النقل؟ وحدة بيانات طبقة الشبكة؟ إطار طبقة ربط البيانات؟
25. أي طبقة من طبقات رصة بروتوكول الإنترنت تتم معالجتها في الموجّه؟ أي طبقة يعالجها محول طبقة الوصلة؟ أي طبقة يعالجها المضيف؟

- **الجزء 1-6**

26. ما الفرق بين الفيروس، والدودة، وحصان طروادة؟
27. صف كيف يمكن إنشاء شبكة روبوت (botnet)، وكيف يمكن استخدامها لعمل هجوم موزّع لحجب الخدمة.
28. افترض أن أليس وبوب يتبادلان الرزم عبر شبكة حاسب. افترض أن ترودي تحشر نفسها في الشبكة في موضع بحيث تستطيع التقاط كل الرزم التي ترسلها أليس وترسل ما تشاء إرساله إلى بوب؛ كما يمكنها أيضاً التقاط كل الرزم التي يرسلها بوب وترسل ما تشاء إرساله إلى أليس. اذكر بعض الأشياء الخبيثة التي يمكن لترودي القيام بها من ذلك الموقع.

❖ تدريبات

1. قم بتصميم ووصف بروتوكول على مستوى التطبيقات يمكن استخدامه بين ماكينة صراف آلي والحاسب المركزي لبنك. يجب أن يسمح بروتوكولك بالتحقق من الرقم السري لبطاقة المستخدم، وبعملية الاستفسار عن الرصيد في الحساب (والذي يحتفظ به الحاسب المركزي)، وكذلك بعملية سحب نقدي (أي صرف النقود للمستخدم). يجب أن تتعامل أجزاء البروتوكول مع الحالة المشهورة عندما لا يتوافر بالماكينة أموال كافية لتغطية السحب المطلوب. حدد البروتوكول بسرد الرسائل التي يتم تبادلها والإجراءات التي يقوم بها كلٌّ من الحاسب المركزي وماكينة الصرف الآلي عند تلقّي تلك الرسائل. وضّح طريقة عمل البروتوكول في حالة عدم وجود أخطاء مستعيناً برسم

يشبه الشكل 1-2. اذكر بالتحديد الافتراض الذي يفترضه البروتوكول بخصوص خدمة النقل التحتية من طرف إلى طرف.

2. خذ في الاعتبار تطبيقاً يرسل البيانات بمعدل ثابت (مثلاً يولد المرسل وحدة بيانات تتألف من N بت كل k وحدة زمن، حيث k رقم صغير وثابت). أيضاً عندما يبدأ مثل ذلك التطبيق فإنه يستمر في العمل لمدة طويلة نسبياً من الوقت. أجب على الأسئلة التالية مع تعليل إجابتك باختصار:

a. ما الذي يناسب هذا التطبيق أكثر، دائرة بتحويل الدوائر أم دائرة بتحويل الرزم؟ لماذا؟

b. افترض أننا استخدمنا شبكة بتحويل الرزم وأن البيانات في تلك الشبكة تأتي فقط من التطبيق المذكور أعلاه. افترض أيضاً أن مجموع معدلات الإرسال الخاصة بالتطبيق أقل من سعات الإرسال لكل وصلة في الشبكة، هل نحن بحاجة إلى تحكم للحد من الازدحام بشكلٍ من الأشكال. لماذا؟

3. خذ في شبكة تحويل الدوائر المبينة في الشكل 1-8. تذكّر أن هناك n دائرة على كل وصلة.

a. ما هو أقصى عدد ممكن من التوصيلات التي يمكن أن تكون شغّالة في نفس الوقت في هذه الشبكة.

c. افترض أن كل التوصيلات هي بين المحوّل في الركن على أقصى اليسار العلوي والمحوّل على الركن في أقصى اليمين السفلي. ما هو الحد الأقصى لعدد التوصيلات التي يمكن أن تكون شغالة في نفس الوقت؟

4. راجع مثال قافلة السيارات في جزء 1-4. مرة أخرى افترض سرعة انتقال مقدارها 100 كم/ساعة.

a. افترض أن القافلة تسير مسافة 200 كم، بدءاً من كشك لتحصيل الرسوم، مروراً بالكشك التالي، وانتهاءاً قبل الكشك الثالث مباشرةً. احسب التأخير من طرف إلى طرف.

b. كرر (a) أعلاه بافتراض أن القافلة تضم 7 سيارات فقط بدلاً من 10 سيارات.

5. نبدأ في هذا التمرين المبسط باستكشاف تأخير الانتقال وتأخير الإرسال، وهما مفهومان أساسيان في مجال الشبكات. خذ في الاعتبار مضيفين A وB موصّلين عبر وصلة واحدة بمعدل إرسال R بت/ثانية. افترض أن المسافة بين المضيفين تبلغ m متر وأن سرعة الانتقال على الوصلة هي s متر/ثانية. يقوم مضيف A بإرسال رزمة طولها L بت إلى مضيف B.

a. عبّر عن تأخير الانتقال d_{prop} بدلالة كلٍّ من m و s.

b. عبّر عن زمن الإرسال d_{trans} بدلالة كلٍّ من L و s.

c. بإهمال تأخير المعالجة وتأخير الانتظار في الصفوف، أوجد تعبيراً رياضياً للتأخير من طرف إلى طرف.

d. افترض أن مضيف A بدأ في إرسال رزمة عند الوقت $t = 0$، أين يكون البت الأخير من الرزمة عند الوقت $t = d_{trans}$.

e. إذا كان d_{prop} أكبر من d_{trans}، فأين يكون البت الأول من الرزمة عند $t = d_{trans}$؟

f. إذا كان d_{prop} أصغر من d_{trans}، فأين يكون البت الأول من الرزمة عند $t = d_{trans}$؟

g. افترض أن $s = 2.5 \times 10^8$ متر/ثانية، $L = 100$ بت، $R = 28$ كيلوبت/ثانية، احسب المسافة m التي تكون عندها $d_{prop} = d_{trans}$.

6. في هذا التمرين سنأخذ في الاعتبار إرسال صوت في الوقت الحقيقي من مضيف A إلى مضيف B عبر شبكة بتحويل الرزم (VoIP). يقوم مضيف A بتحويل إشارة الصوت التناظرية مباشرةً إلى سلسلة بتات رقمية بمعدل إرسال 64 كيلوبت/ثانية. يقوم مضيف A بعد ذلك بتجميع تلك البتات على شكل رزم طول كلٍّ منها 48 بايت. يوجد وصلة واحدة بين المضيفين A وB لها معدل إرسال قدره 1 ميجابت/ثانية وتأخير انتقال قدره 2 ميللي ثانية. بمجرد تجميع مضيف A لرزمة كاملة، فإنه يقوم بإرسالها إلى مضيف B. بمجرد استلام مضيف B لرزمة كاملة، يقوم بتحويل بتات الرزمة إلى إشارة تناظرية. كم من الزمن ينقضي بين توليد بت (من الإشارة التناظرية الأصلية عند مضيف A) إلى أن يُفك تكويد تلك البت (كجزء من الإشارة التناظرية التي يتم استعادتها عند مضيف B)؟

7. افترض أن عدة مستخدمين يشتركون فيما بينهم في وصلة معدل إرسالها 1 ميجابت/ثانية. افترض أيضاً أن كل مستخدم يحتاج إلى 100 كيلوبت/ثانية عندما يرسل، غير أن كل مستخدم يرسل فقط أثناء 10٪ من الوقت (انظر المناقشة حول الإرسال المتعدد الإحصائي في الجزء 1-3).

a. عند استخدام تحويل الدوائر، كم عدد المستخدمين الذين سيكون من الممكن استيعابهم؟

b. في الجزء المتبقي من هذا التمرين، افترض أننا نستخدم تحويل الرزم. أوجد احتمال أن يقوم مستخدم بعينه بالإرسال.

c. افترض أن هناك 40 مستخدماً. أوجد احتمال أن يقوم n مستخدماً بالضبط بالإرسال معاً في نفس الوقت. (ملاحظة: استخدم التوزيع الإحصائي ذي الحدين (binomial distribution).

d. احسب احتمال أن يوجد 11 مستخدماً أو أكثر يقومون بالإرسال معاً.

8. خذ في الاعتبار المناقشة في جزء 1-3 حول الإرسال المتعدد الإحصائي، حيث أوردنا مثالاً لوصلة سعة إرسالها 1 ميجابت/ثانية. يقوم كلٌّ من المستخدمين بتوليد البيانات بمعدل 100 كيلوبت/ثانية في حالة نشاطه ولكنه ينشط باحتمال قدره $p = 0.1$. افترض أن الوصلة بسعة 1 ميجابت/ثانية تم استبدالها بوصلة سعتها 1 جيجابت/ثانية.
   a. احسب أقصى عدد يمكن استيعابه من المستخدمين في نفس الوقت باستخدام تحويل الدوائر (N).
   b. الآن خذ في الاعتبار تحويل الرزم وعدد M مستخدم. أوجد معادلة (في p وM وN) لاحتمال أن يكون هناك أكثر من N مستخدماً يقومون بالإرسال في نفس الوقت.
9. خذ في الاعتبار رزمة طولها L بتاً تبدأ على نظام طرفي، وتنتقل على وصلة واحدة إلى محوّل رزم، ثم تنتقل من محوّل الرزم على وصلة ثانية إلى وجهتها النهائية. افترض أن d_i وs_i و R_i تمثل (من اليمين إلى اليسار) الطول، وسرعة الانتقال، ومعدل الإرسال للوصلة i، حيث $i = 1, 2$. يتسبب محوّل الرزم في تأخير قدره d_{proc} لكل رزمة. بإهمال تأخير الانتظار في الصفوف، أوجد التأخير الكلي للرزمة من طرف إلى طرف بدلالة L و d_i وs_i وR_i $(i = 1, 2)$. افترض الآن أن الرزمة طولها 1000 بايت، وسرعة الانتقال على كلا الوصلتين 2.5×10^8 متر/ثانية، ومعدل الإرسال على كلا الوصلتين 1 ميجابت/ثانية، وتأخير المعالجة في محوّل الرزم هو 1 ميللي ثانية، وطول الوصلة الأولى 4000 كم، وطول الوصلة الثانية 1000 كم. احسب التأخير من طرف إلى طرف في هذه الحالة.
10. في التمرين أعلاه، افترض أن $R = R_1 = R_2$ و$d_{proc} = 0$. افترض أيضاً أن محوّل الرزم لا يتبع أسلوب التخزين ثم الإرسال للرزم، ولكن بدلاً من ذلك يقوم في الحال بإرسال كل بت يستلمه دون الانتظار حتى يستلم بقية الرزمة بأكملها. احسب التأخير من طرف إلى طرف في هذه الحالة.
11. يقوم محوّل الرزم باستلام رزمة وتحديد أي الوصلات الخارجة منه ينبغي تمرير الرزمة إليها. عند وصول الرزمة كانت هناك رزمة أخرى تم إرسال نصفها بالفعل على الوصلة الخارجة المطلوبة، بالإضافة إلى 3 رزم أخرى تنتظر دورها في الإرسال. يتم إرسال الرزم بترتيب وصولها. افترض أن الرزم كلها لها نفس الطول وقدره 1000 بايت وأن معدل الإرسال على الوصلة هو 1 ميجابت/ثانية. احسب تأخير الانتظار في الصف للرزمة الواصلة. بشكلٍ أكثر عموماً، ما هو تأخير الانتظار في الصف عندما تكون كل الرزم بطول L، ومعدل الإرسال R، وتم إرسال x بت من الرزمة الجاري إرسالها حالياً، وهناك n رزمة تنتظر في الصف دورها في الإرسال؟

12. افترض أن N رزمة تصل معاً إلى وصلة لا يوجد عليها رزم أخرى يجري إرسالها أو تنتظر دورها في الإرسال. كل رزمة طولها L ومعدل الإرسال على الوصلة هو R . ما هو متوسط تأخير الانتظار في الصف لـ N رزمة؟
13. خذ في الاعتبار تأخير الانتظار في الصف في المخزن المؤقت على موجّه (قبل الوصلة الخارجة). افترض أن كل رزمة طولها L ومعدل الإرسال على الوصلة هو R وأن N رزمة تصل معاً إلى المخزن المؤقت كل LN/R ثانية. أوجد متوسط تأخير الانتظار في الصف لرزمة. (ملاحظة: تأخير الانتظار في الصف للرزمة الأولى هو صفر، وللثانية L/R، وللثالثة $2L/R$. تكون الرزمة رقم N قد أُرسلت بالفعل عند وصول الدفعة الثانية من الرزم.
14. خذ في الاعتبار تأخير الانتظار في الصف في المخزن المؤقت على موجّه. افترض أن I تمثل كثافة حركة مرور البيانات، أي $I = La/R$. افترض أن تأخير الانتظار في الصف يُعبّر عنه بالمعادلة $IL/R(1- I)$ حيث $I < 1$.
    a. أوجد معادلة للتأخير الكلي، أي تأخير الانتظار في الصف بالإضافة إلى تأخير الإرسال.
    b. ارسم التأخير الكلي كدالة في L/R.
15. a. قم بتعميم معادلة التأخير من طرف إلى طرف في الجزء 1-4-3 لحالة التأخيرات غير المتجانسة للمعالجة، والإرسال، والانتقال.
    b. كرر (a)، ولكن مع افتراض وجود تأخير انتظار متوسط عند كل عقدة قدره d_{queue}.
16. استخدم برنامج تتبع المسار (Traceroute) بين مصدر ووجهة على نفس القارة عند 3 ساعات مختلفة من اليوم.
    a. احسب المتوسط والانحراف المعياري للوقت الذي تستغرقه رحلة الذهاب والعودة في كلٍّ من الساعات الثلاث.
    b. أوجد عدد الموجّهات على المسار في كلٍّ من الساعات الثلاث. هل يختلف المسار من ساعة إلى أخرى؟
    c. حاول التعرف على شبكات موفري خدمة الإنترنت (ISP) التي تعبرها الرزم من المصدر إلى الوجهة. الموجهات التي تحمل أسماء أو عناوين IP متشابهة يمكن اعتبارها تابعة لنفس موفر خدمة الإنترنت. في تجاربك هذه، هل تحدث أكبر التأخيرات عند الواجهات النظيرة التي تصل مابين موفري خدمة انترنت متجاورين؟
    d. كرر التجربة أعلاه لمصدر ووجهة على قارتين مختلفتين. قارن بين النتائج على نفس القارة والنتائج عبر قارتين.

17. خذ في الاعتبار مثال الطاقة الانتاجية المناظر للشكل 1-16 (b). افترض الآن أن هناك M زوج من الزبائن/الخادمات بدلاً من 10 أزواج. افترض أن R_s و R_c وR تمثل (من اليمين إلى اليسار) معدلات الإرسال على وصلات الخادمات، وصلات الزبائن، ووصلة الشبكة، على الترتيب. افترض أن كل الوصلات الأخرى لها سعات إرسال كافية وأنه لا توجد حركة مرور بيانات داخل الشبكة غير حركة المرور التي تولدها أزواج الزبائن والخادمات أعلاه. قم باشتقاق تعبير رياضي للطاقة الإنتاجية بدلالة R_s و R_c و R و M.
18. افترض مضيفين A وB تفصلهما مسافة 10000 كم ويتصلان عبر وصلة مباشرة معدل الإرسال عليها 1 ميجابت/ثانية. افترض أن سرعة الانتقال على تلك الوصلة هي 2.5×10^8 متر/ثانية.
    a. احسب حاصل ضرب سعة الإرسال (الحيز الترددي) × التأخير، أي $R\times d_{prop}$.
    b. خذ في الاعتبار إرسال ملف حجمه 400000 بت من مضيف A إلى مضيف B. وافترض أن الملف أُرسل دفعة واحدة كرسالة واحدة كبيرة. ما هو العدد الأقصى من البتات التي تكون موجودة على الوصلة في أي وقت؟
    c. اذكر تفسيراً لمفهوم حاصل الضرب سعة الإرسال × التأخير.
    d. ما هو عرض البت (بالمتر) على الوصلة؟ هل هي أطول من ملعب كرة قدم؟
    e. قم باشتقاق تعبير رياضي عام لعرض البت بدلالة سرعة الانتقال s، ومعدل الإرسال R، وطول الوصلة m.
19. بالرجوع إلى التمرين 18، افترض أننا يمكننا تعديل R. ما هي قيمة R التي تجعل عرض البت على الوصلة مساوياً لطول الوصلة نفسها؟
20. خذ في الاعتبار التمرين 18 ولكن بوصلة لها معدل إرسال R قيمته 1 جيجابت/ثانية
    a. احسب حاصل ضرب سعة الإرسال (الحيز الترددي) × التأخير، أي $R \times d_{prop}$.
    b. خذ في الاعتبار إرسال ملف حجمه 400000 بت من مضيف A إلى مضيف B. افترض أن الملف أُرسل دفعة واحدة كرسالة واحدة كبيرة. ما هو العدد الأقصى من البتات التي تكون موجودة على الوصلة في أي وقت؟
    c. ما هو عرض البت (بالمتر) على الوصلة؟
21. بالرجوع إلى التمرين 18 مرة أخرى.
    a. كم يستغرق إرسال الملف، بافتراض أنه يُرسَل دفعة واحدة؟
    b. افترض الآن أن الملف يتم تجزيئه إلى 10 رزم يضم كلٌّ منها 40000 بت. افترض أن المستقبل يرسل إشعاراً باستلام كل رزمة وأن فترة إرسال رزمة إشعار الاستلام يمكن إهمالها. أخيراً افترض أن المرسل لايمكنه إرسال الرزمة التالية قبل أن يتلقى إشعاراً باستلام الرزمة التي أرسلها. كم يستغرق إرسال الملف في هذه الحالة؟

c. قارن بين النتائج في كلٍّ من الحالتين (a) و(b).

22. خذ في الاعتبار وصلة ميكروويف بمعدل إرسال 10 ميجابت/ثانية بين قمر صناعي ثابت بالنسبة للكرة الأرضية ومحطة القاعدة له على الأرض. في كل دقيقة يأخذ القمر صورة رقمية ويرسلها إلى محطة القاعدة. افترض أن سرعة الانتقال هي 2.4×10^8 متر/ثانية.

a. ما هو تأخير الانتقال على الوصلة؟

b. احسب حاصل ضرب سعة الإرسال (الحيز الترددي) x التأخير، أي $d_{\text{prop}} \times R$.

c. افترض أن x تمثل حجم الصورة بالبتات. ما هو الحد الأدنى لقيمة x بحيث تبقى وصلة الميكروويف تعمل باستمرار؟

23. خذ في الاعتبار مثال السفر على شركات الطيران الوارد ضمن مناقشتنا لطبقية البروتوكولات في الجزء 1-5، وكذلك إضافة الترويسات إلى وحدات بيانات البروتوكولات أثناء نزولها من أعلى رصة البروتوكول إلى أسفلها لدى المرسل. هل هناك ما يناظر ذلك من أشياء تضاف إلى الركاب وأمتعتهم عند انتقالهم من أعلى رصة بروتوكول شركة الطيران إلى أسفلها؟.

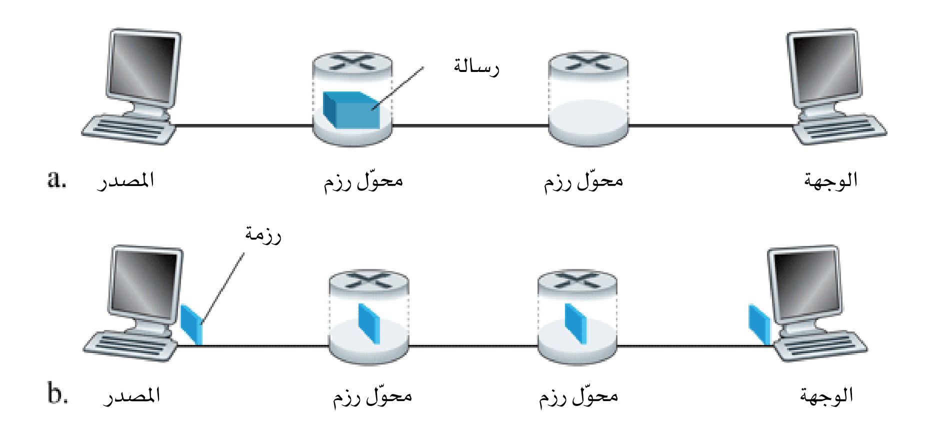
24. في الشبكات الحديثة لتحويل الرزم يقوم المضيف بتجزئة الرسائل الكبيرة من طبقة التطبيقات (مثلاً صورة أو ملف صوتي) إلى رزم صغيرة يرسلها عبر الشبكة. يقوم المستقبل بعد ذلك بتجميع الرزم الواصلة للحصول على الرسالة الأصلية. نطلق على هذه العملية تجزئة الرسائل. يوضح الشكل 1-24 إرسال رسالة من طرف إلى طرف بدون تجزئة الرسائل وبتجزئة الرسائل. افترض أننا نود إرسال رسالة طولها 7.5×10^6 بت من المصدر إلى الوجهة في الشكل 1-24. افترض أن معدل الإرسال هو 1.5 ميجابت/ ثانية على كل الوصلات، واهمل تأخيرات الانتقال والانتظار في الصف والمعالجة.

a. خذ في الاعتبار إرسال الرسالة من المصدر إلى الوجهة **بدون** تجزئة. كم يستغرق من الوقت نقل الرسالة من مضيف المصدر إلى محوّل الرزم الأول؟ مع مراعاة أن كل محوّل يستخدم أسلوب التخزين ثم الإرسال بتحويل الرزم، ما هو الوقت الكلي اللازم لنقل الرسالة من مضيف المصدر إلى مضيف الوجهة؟

b. افترض الآن أن الرسالة يتم تجزئتها إلى 5000 رزمة طول كلٍّ منها 1500 بت. كم يستغرق من الوقت نقل الرزمة الأولى من مضيف المصدر إلى محوّل الرزم الأول؟ بينما يتم نقل الرزمة الأولى من محوّل الرزم الأول إلى محوّل الرزم الثاني، يتم نقل الرزمة الثانية من مضيف المصدر إلى محوّل الرزم الأول. في أي وقت سيتم استلام الرزمة الثانية بكاملها عند محوّل الرزم الأول؟

c. كم يستغرق من الوقت نقل الملف من مضيف المصدر إلى مضيف الوجهة مع تجزئة الرسالة؟ قارن هذه الإجابة مع إجابتك في الجزء (a) وعلق على النتيجة.

d. اذكر عيوب تجزئة الرسالة.



الشكل 1-24 نقل الرسائل من طرف إلى طرف: (a) بدون تجزئة الرسائل؛ (b) بتجزئة الرسائل.

25. قم بتجريب برنامج جافا الخاص بتجزئة الرسالة (Message Segmentation Applet) على موقع الويب المصاحب لهذا الكتاب. هل تتوافق التأخيرات التي يحسبها البرنامج مع نتائج التأخيرات في المسألة السابقة؟ كيف تؤثر تأخيرات الانتقال عبر الوصلات على التأخير من طرف إلى طرف باستخدام تحويل الرزم (أي مع تجزئة الرسالة) وباستخدام تحويل الرسائل (أي بدون تجزئة الرسالة).

26. خذ في الاعتبار عملية إرسال ملف يتألف من F بت من مضيف A إلى مضيف B. يوجد وصلتان (ومحوّل واحد) بين A وB، والوصلتان غير مزحومتين (أي لا توجد تأخيرات انتظار). يقوم مضيف A بتجزئة الملف إلى قطع طول كلٍّ منها S بت ويضيف ترويسة طولها 40 بتاً لكل قطعة، ومن ثم يكوّن رزماً طول كلٍّ منها $L = 40 + S$ بت. معدل الإرسال على كل وصلة هو R بت/ثانية. أوجد قيمة S التي تعطي الحد الأدنى للتأخير في نقل الملف من مضيف A إلى مضيف B. اهمل تأخير الانتقال.

❖ أسئلة مناقشة

1. أي أنواع خدمات الوصول اللاسلكي تتوافر في منطقتك؟
2. باستخدام تقنية 802.11 للشبكات المحلية اللاسلكية، قم بتصميم شبكة منزلية لمنزلك أو منزل عائلتك. قم بإعداد قائمة بموديلات الأجزاء المطلوبة بالتحديد مع كلفة كلٍّ منها.
3. صف خدمة سكايب من حاسب شخصي إلى حاسب شخصي (PC-to-PC). قم بتجريب خدمة سكايب للفيديو بين حاسبين شخصيين واكتب تقريراً مختصراً عن تجربتك.
4. توفر سكايب خدمة تتيح لك عمل مكالمة هاتفية من حاسب شخصي إلى هاتف عادي. هذا يعني أن المكالمة تمر عبر كلٍّ من الإنترنت وشبكة الهاتف. اشرح كيف يمكن تنفيذ ذلك.
5. ما هي خدمة الرسائل القصيرة (SMS)؟ في أي البلدان/القارات تنتشر تلك الخدمة؟ هل يمكن إرسال رسالة SMS من موقع على الويب إلى هاتف نقال؟
6. ما هو تشغيل الفيديو المخزّن؟ اذكر بعض المواقع الشهيرة على الويب التي توفر تلك الخدمة حالياً.
7. ما هو تشغيل الفيديو الحي عن طريق مشاركة النظائر للملفات (P2P). اذكر بعض المواقع التي توفر تلك الخدمة حالياً.
8. قم بتحديد 5 شركات توفر خدمات مشاركة النظائر للملفات (P2P). لكل شركة من الشركات اذكر نوع محتويات الملفات التي يمكنها التعامل معها.
9. من اخترع خدمة ICQ (أول خدمة للرسائل الفورية)؟ متى كان ذلك؟ وكم كان عمر المخترع وقتها؟ أيضاً من اخترع Napster؟ متى كان ذلك؟ وكم كان عمر المخترع وقتها؟
10. قارن بين وصول WiFi اللاسلكي للإنترنت ووصول 3G اللاسلكي للإنترنت. ماهي معدلات الإرسال لكلٍّ من الخدمتين؟ ماهي الكلفة؟ قم بمناقشة موضوع التجوال وتوافر الخدمة في كل وقت وفي كل مكان.
11. لماذا لم تعد خدمة Napster الأصلية لمشاركة النظائر للملفات موجودة الآن؟ ماهي منظمة RIAA وماهي الإجراءات التي تتخذها للحد من مشاركة النظائر للملفات التي تخضع محتوياتها لحقوق ملكية؟ مالفرق بين خرق حقوق الملكية بشكلٍ مباشر وبشكلٍ غير مباشر؟
12. ما هو نظام BitTorrent لتوزيع الملفات؟ وكيف يختلف بصورة جوهرية عن خدمات مشاركة النظائر للملفات (P2P) مثل eDonkey و LimeWire و Kazaa؟

13. هل تتوقع أنه بعد 10 سنوات من الآن ستكون المشاركة في ملفات لها حقوق ملكية منتشرةً على نطاق واسع على شبكات الحاسب؟ علل إجابتك.

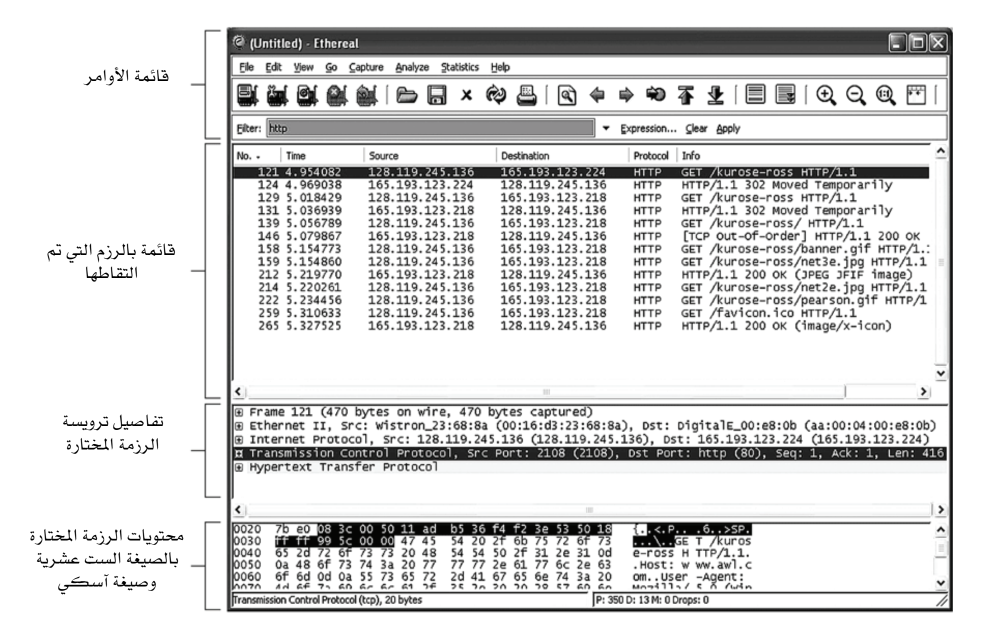
❖ مختبر إيثيريل

"قل لي وسأنسى، أرني وسأتذكر، أشركني وسأفهم" – مَثَلٌ صيني

يمكن تعميق فهم بروتوكولات الشبكات في أحيان كثيرة عن طريق مشاهدة تلك البروتوكولات وهي تعمل والتجريب معها – بملاحظة الرسائل التي يتبادلها بروتوكولان، والدخول في تفاصيل عمل البروتوكولات، وجعلها تقوم بإجراءات معينة ثم ملاحظة تلك الإجراءات والنتائج المترتبة عليها. يمكن القيام بذلك من خلال سيناريوهات يتم محاكاتها أو في بيئة شبكة حقيقية كالإنترنت. تستخدم برامج جافا التفاعلية الموجودة في موقع الويب المصاحب لهذا الكتاب الطريقة الأولى. في مختبر إيثيريل سنتبع الطريقة الثانية. ستقوم أنت بتشغيل تطبيقات شبكات في سيناريوهات مختلفة باستخدام حاسب على مكتبك أو في المنزل أو في مختبر. ستراقب بروتوكولات الشبكة على حاسبك وهي تتفاعل وتتبادل الرسائل مع كيانات بروتوكولات أخرى تعمل في أماكن أخرى على الإنترنت. ستكون أنت وحاسبك جزءاً من تلك الاختبارات الحية. ستلاحظ – وستتعلم – عن طريق عمل الأشياء بنفسك.

يطلق على الأداة الأساسية لمراقبة الرسائل التي يتم تبادلها بين كيانات البروتوكول اسم لاقط الرزم (Packet Sniffer). كما يوحي الاسم، يقوم لاقط الرزم بشكلٍ سلبي بنسخ (التقاط) الرزم التي يرسلها أو يستقبلها حاسبك، كما يعرض محتويات مختلف حقول البروتوكول لتلك الرزم. يوضح الشكل 1-25 لقطة شاشة من برنامج إيثيريل لالتقاط الرزم. هذا البرنامج مجاني ويعمل على الحاسبات بنظم التشغيل ويندوز، ولينكس/يونيكس، وماكنتوش. خلال هذا الكتاب ستجد تمرينات مختبر إيثيريل تمكّنك من استكشاف عدد من البروتوكولات التي تناولناها. في أول مختبر إيثيريل ستحصل على نسخة من البرنامج وتقوم بتثبيته على حاسبك، ثم تتصل بموقع ويب وتلتقط وتفحص رسائل بروتوكولات يتم تبادلها بين متصفح الويب على حاسبك وخادم الويب لذلك الموقع.

يمكنك الحصول على كافة التفاصيل عن مختبر إيثيريل الأول (بما في ذلك تعليمات عن كيفية الحصول على برنامج إيثيريل وتثبيته) من موقع الويب المصاحب للكتاب http://www.aw1.com/kurose-ross.



الشكل 1-25 لقطة شاشة من برنامج إيثيريل

طبقة التطبيقات

Application Layer

محتويات الفصل:

- مبادئ تطبيقات الشبكة
- شبكة الويب وبروتوكول HTTP
- نقل الملفات باستخدام بروتوكول FTP
- البريد الإلكتروني (E-mail)
- خدمة دليل الإنترنت لأسماء النطاقات (DNS)
- تطبيقات النظائر (P2P)
- برمجة مقابس بروتوكول TCP
- برمجة مقابس بروتوكول UDP
- الخلاصة

تعتبر تطبيقات الشبكة المبرر لوجود شبكة الحاسب الآلي، فإذا لم نستطع تخيل أية تطبيقات مفيدة لتلك الشبكة، فلن تكون هناك حاجة لتصميم أي بروتوكولات لدعمها. خلال الأربعين سنة الماضية تم ابتكار العديد من التطبيقات المبدعة والرائعة للشبكة. يشمل ذلك التطبيقات التقليدية النصية والتي كانت شائعة في السبعينيات والثمانينيات، كالبريد الإلكتروني النصي (text e-mail)، والوصول إلى الحاسبات عن بُعد (remote login)، ونقل الملفات (file transfer)، ومجموعات الأخبار (newsgroups)، والدردشة النصية (text chatting). كما تضمنت ذلك التطبيق الهام – الويب Web – الذى ظهر في منتصف التسعينيات مما يسّر المشاركة في المعلومات والبحث عنها والتجارة الإلكترونية. وتضمنت أيضاً التطبيقين الهامين اللذين ظهرا في نهاية الألفية: الرسائل الفورية (instant messaging) بقوائم المراسلة، ومشاركة النظائر للملفات (P2P file sharing). بالإضافة إلى العديد من تطبيقات الوسائط المتعددة (multimedia) كعرض شرائط الفيديو وراديو الإنترنت وهاتف الإنترنت ومؤتمرات الفيديو وتليفزيون الإنترنت (IPTV). وعلاوة على ذلك فإن انتشار تقنية الوصول ذات الحيز الترددي العريض للأماكن السكنية والاستخدام المتزايد للشبكات اللاسلكية يهيئ المسرح للمزيد من التطبيقات الجديدة والمثيرة في المستقبل.

في هذا الفصل سندرس جوانب تطوير تطبيقات الشبكة من حيث المفهوم والتطبيق. في البداية سنُعرِّف المفاهيم الرئيسة لطبقة التطبيقات (البرامج)، بما في ذلك بروتوكولات طبقة التطبيقات والزبائن (clients) والخادمات (servers) والعمليات (processes) ومقابس الاتصال (sockets) وواجهات التعامل (interfaces) مع طبقة النقل. بعد ذلك سنستعرض عدة أمثلة لتطبيقات الشبكة بالتفصيل كالويب والبريد الإلكتروني وخدمة الدليل لأسماء النطاقات (DNS) ومشاركة النظائر للملفات (P2P) وهاتف الإنترنت عن طريق النظائر. نتناول بعد ذلك تطوير تطبيقات للشبكة على كلٍّ من بروتوكول TCP وبروتوكول UDP. سندرس على التحديد واجهة المقابس لبرمجة التطبيقات (API)، ونتتبع تطبيقات بسيطة للزبون والخادم بلغة جافا (Java). وكمثال سندرس كيف يمكن تطوير خادم ويب

بسيط بتلك اللغة، كما سنقدم أيضاً عدة تدريبات مثيرة ومسلية على برمجة المقابس في نهاية الفصل.

تعتبر طبقة التطبيقات بصفة خاصة مكاناً جيداً لبدء دراسة البروتوكولات، فهي مألوفة لنا من خلال تعاملنا مع العديد من التطبيقات (البرامج) التي تعتمد على البروتوكولات التي سندرسها. هذا سيعطينا فكرةً جيدةً عن ماهية البروتوكولات ومدخلاً للعديد من القضايا التي ستقابلنا مرة أخرى عندما ندرس بروتوكولات طبقة النقل وطبقة الشبكة وطبقة ربط البيانات.

1-2 مبادئ تطبيقات الشبكة

لنفرض أن لديك فكرة لتطبيق جديد؛ قد يكون خدمة عظيمة للإنسانية أو لنيل مدح أستاذك أو لنيل ثروة عظيمة أو ببساطة قد يكون للمرح والتسلية. مهما يكن الحافز دعنا نستعرض الآن كيف تُحوِّل تلك الفكرة إلى تطبيق حقيقي للشبكة.

من صميم تطوير تطبيقات للشبكة كتابة البرامج التي تُنفَّذ على الأنظمة الطرفية (end systems) المختلفة وتتصل فيما بينها عبر الشبكة. على سبيل المثال في تطبيق الويب هناك برنامجان مُميَّزان يتصلان مع بعضهما: برنامج المتصفّح، وبرنامج خادم الويب. يعمل برنامج المتصفّح على مضيف المُستخدِم (مثلاً حاسب مكتبي (desktop) أو حاسب نقّال (laptop) أو مساعد رقمي شخصي (PDA) أو هاتف جوال (mobile phone) أو ما شابه ذلك)، ويعمل برنامج خادم الويب على مضيف خادم الويب. وكمثال آخر في نظام مشاركة النظائر للملفات يوجد برنامج في كل مضيف في المجموعة المشاركة للملفات، وفي هذه الحالة قد تكون البرامج في المضيفات المختلفة متشابهة أو متطابقة.

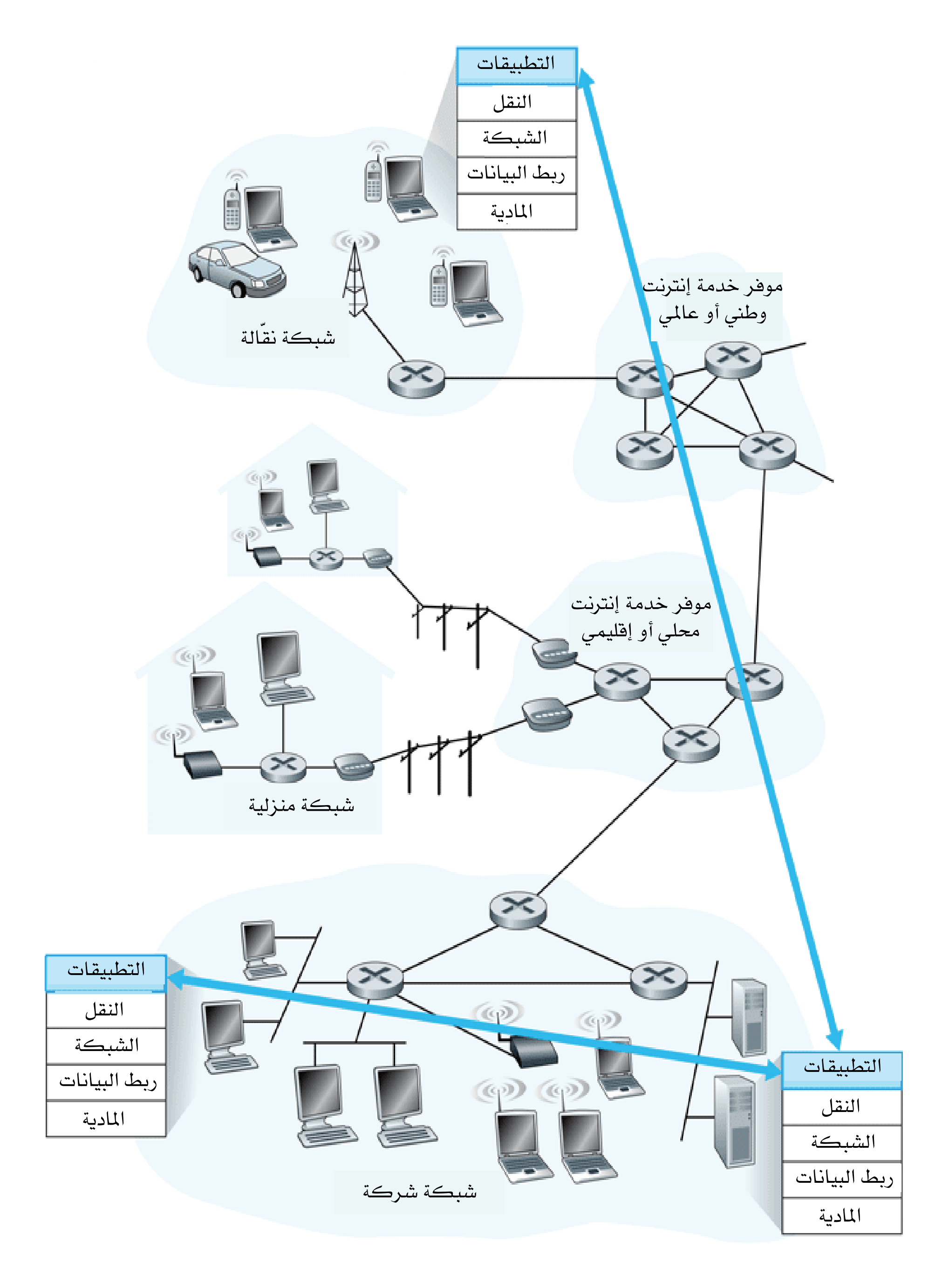
وهكذا فعند تطوير تطبيقك الجديد، ستحتاج إلى كتابة البرامج التي سيتم تنفيذها على أنظمة طرفية متعددة. ويمكنك كتابة هذه البرامج على سبيل

المثال بلغة C أو ++C أو جافا. المهم أنك لست بحاجة إلى أن تكتب برامج للأجهزة التي في قلب الشبكة (network core devices)، كالموجّهات (routers) أو محوّلات طبقة ربط البيانات (switches). وحتى إذا أردت كتابة برامج تطبيقات لتلك الأجهزة، فلن يكون بوسعك عمل ذلك. فكما تعلمنا في الفصل الأول (انظر الشكل 1-20) لا تعمل الأجهزة الموجودة في قلب الشبكة في طبقة التطبيقات، ولكنها تعمل في الطبقات الدنيا وتحديداً في طبقة الشبكة وما تحتها. لقد سهّل هذا التصميم البسيط – أي حصر برامج التطبيقات في الأنظمة الطرفية كما هو موضح في الشكل 2-1 – تطوير وانتشار تشكيلة واسعة من تطبيقات الشبكة بسرعة.

2-1-1 البنية المعمارية لتطبيقات الشبكة

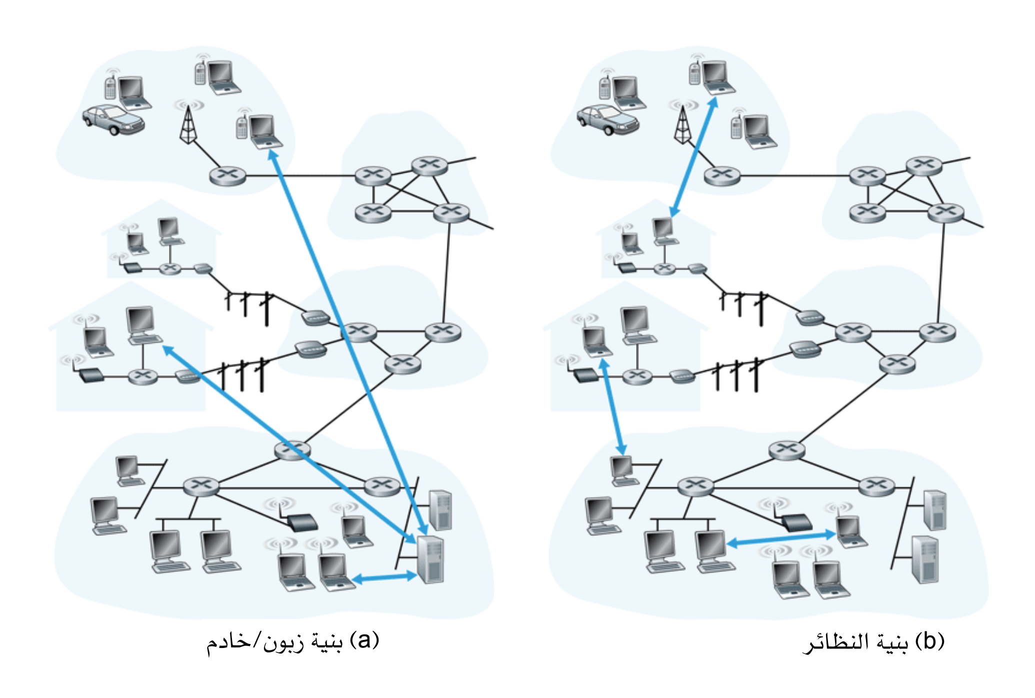
قبل الخوض في كتابة البرامج يجب أن يكون لديك خطة معمارية شاملة لبنية تطبيقك. تذكّر أن بنية التطبيق مختلفة تماماً عن بنية الشبكة (أي نموذج الإنترنت الذى يتكون من خمس طبقات كما تناولناه في الفصل الأول على سبيل المثال)، ومن منظور مطوِّر التطبيقات تعتبر بنية الشبكة ثابتة، وتستخدم لتوفير مجموعة معينة من الخدمات للتطبيقات. ومن ناحية أخرى يتم تصميم بنية التطبيق بواسطة المطوِّر والذي يحدد كيفية هيكلة التطبيق على الأنظمة الطرفية المختلفة. عند تحديد بنية التطبيق المعمارية، سيختار مطوِّر التطبيقات على الأرجح أحد النموذجين السائدين والمُستخدَمين في تطبيقات الشبكة الحديثة: بنية زبون/خادم أو بنية النظائر.

في بنية زبون/خادم يوجد مضيف يعمل على الدوام يسمى "الخادم" ليلبي طلبات العديد من المضيفات الأخرى تسمى "الزبائن". يمكن لمضيفات الزبائن أن تكون شغّالة أحياناً أو دائماً. المثال التقليدى لذلك هو تطبيق الويب الذى يعمل فيه خادم الويب بشكلٍ دائم في خدمة طلبات برامج المتصفّحات التي تعمل على مضيفات الزبائن. وعندما يتلقى خادم الويب طلباً لشيءٍ (كائن object) ما (مثل صفحة ويب) من مضيف الزبون، فإنه يستجيب بإرسال ذلك الشيء إليه إذا كان



الشكل 2-1 يحدث الاتصال بين برامج الشبكة بين النظم الطرفية في طبقة التطبيقات.

لدى الخادم نسخة منه، أو يستجيب بإرسال رسالة خطأ إذا لم يكن موجوداً لديه. لاحظ أنه في بنية زبون/خادم لا يتصل الزبائن مباشرة ببعضهم البعض، فمثلاً في تطبيق الويب لا يتصل متصفّحان مباشرة. الخاصية الأخرى لبنية زبون/خادم هي أن للخادم عنواناً ثابتاً ومعروفاً، يُعرَف بعنوان IP (سنتناوله فيما بعد). وبما أن للخادم عنواناً ثابتاً معروفاً وبما أنه يعمل باستمرار، يمكن أن يتصل الزبون بالخادم دائماً (في أي وقت) بإرسال رزمة بيانات إلى عنوان الخادم. من التطبيقات المعروفة جيداً التي تستخدم بنية زبون/خادم: تطبيقات الويب (Web)، ونقل الملفات (FTP)، والوصول إلى الحاسبات عن بُعد (Telnet)، والبريد الإلكتروني (e-mail). يوضح الشكل 2-2 (a) بنية زبون/خادم.



الشكل 2-2 بنية التطبيقات: (a) بنية زبون/خادم، (b) بنية النظائر.

في أغلب الأحيان في تطبيقات زبون/خادم قد يعجز مضيف خادم واحد عن تلبية كل طلبات الزبائن. على سبيل المثال يمكن أن يزدحم موقع مشهور للتواصل الاجتماعي بسرعة إذا كان يستخدم خادماً واحداً لمعالجة كل الطلبات. لهذا السبب غالباً ما تُستخدَم مجموعة من الخادمات (cluster of servers)، يطلق عليها مزرعة خادمات (server farm)، للحصول على خادم افتراضي قوي في بنية زبون/خادم. غالباً ما تكون خدمات التطبيقات التي تعتمد أسلوب زبون/خادم مرتكزة بدرجة كبيرة على بنية تحتية (infrastructure)، لأنها تتطلب من موفّري الخدمة شراء وتركيب وصيانة مزارع الخادمات. إضافةً إلى ذلك فإن موفّري الخدمة يجب أن يدفعوا تكلفة متكررة للاتصال وللحيّز الترددي اللازمين لإرسال البيانات إلى الإنترنت وتلقيها منها. يلاحظ أن الخدمات الشائعة مثل محركات البحث (كـ Google مثلاً)، والتجارة من خلال الإنترنت (كـ Amazon وeBay)، والبريد الإلكتروني من خلال الويب (كبريد Yahoo)، والتشبيك الاجتماعي (كـ MySpace وFacebook)، ومشاركة الفيديو (كـ YouTube) ترتكز بدرجة كبيرة على بنية تحتية ويتطلب توفيرها كلفة عالية.

وعلى النقيض من ذلك تعتمد بنية النظائر بدرجة بسيطة على خادمات البنية التحتية التي تعمل دائماً أو قد لا تحتاج إليها على الإطلاق. وكبديل لذلك يستخدم التطبيق الاتصال المباشر بين أزواج المضيفات (والتي يطلق عليها النظائر) بشكلٍ متقطع. وجدير بالذكر أن تلك النظائر ليست ملكاً لموفّر الخدمة، ولكنها أجهزة حاسبات مكتبية وحاسبات نقّالة تحت سيطرة المستخدمين، حيث يوجد معظم النظائر في البيوت والجامعات والمكاتب. ولما كانت النظائر تتصل فيما بينها دون الحاجة للمرور عبر خادم مخصّص، فإنه يُطلق على تلك البنية "نظير إلى نظير" (peer-to-peer). يعتمد العديد من التطبيقات الأكثر شعبية والمزدحمة بحركة البيانات اليوم على بنية النظائر، بما في ذلك توزيع الملفات (كـ BitTorrent)، والمشاركة والبحث عن الملفات (كـ eMule وLimeWire)، وهاتف الإنترنت (كـ Skype)، وتليفزيون الإنترنت (كـ PPLive). يوضح الشكل 2-2 (b) بنية النظائر. يلاحظ أن لبعض التطبيقات بنية معمارية هجينة (hybrid) تجمع

عناصر من كلٍّ من بنية زبون/خادم وبنية النظائر. على سبيل المثال هناك العديد من تطبيقات الرسائل الفورية تستعمل خادمات لتعقب عناوين IP للمُستخدِمين، لكن ترسل الرسائل من مُستخدِمٍ إلى آخر مباشرة بين المضيفات (بدون المرور عبر خادمات وسيطة).

من أهم مزايا بنية النظائر قدرتها الذاتية على التوسع (scalability). على سبيل المثال في تطبيق مشاركة الملفات بتلك البنية، رغم أن كل نظير يولِّد حمل شغل (workload) إضافياً بطلب ملفات، فإن كل نظير يضيف مزيداً من قدرة الخدمة أيضاً إلى النظام بقيامه بتوزيع الملفات إلى النظائر الأخرى. تعتبر بنية النظائر طريقة فعالة من حيث التكلفة، فهي لا تتطلب عادةً بنية خادمات تحتية كبيرة أو حيّزاً ترددياً كبيراً للخادم. في سعيهم لخفض كلفة التشغيل، يزداد اهتمام موفّري الخدمة (مثل MSN، وYahoo، وغيرهما) باستخدام بنية النظائر لتطبيقاتهم. ولكن من ناحية أخرى وبسبب الطبيعة الموزّعة والمفتوحة جداً لتطبيقات بنية النظائر، يكون من الصعب توفير نظام فعّال لتأمينها [Doucer 2002; Yu 2006; Liang 2006; Naoumov 2006].

2-1-2 العمليات المُتصِلة فيما بينها (Communicating Processes)

قبل بناء تطبيق للشبكة تحتاج أيضاً لفهم أساسي لكيفية اتصال البرامج – والتي تعمل في أنظمة طرفية متعددة – مع بعضها البعض. في المصطلحات التخصصية لنظم التشغيل ليست البرامج في حقيقة الأمر هي التي تتصل فيما بينها ولكنها العمليات (processes) (ويمكن أن تعتبر العملية كبرنامج يعمل ضمن نظام طرفي). عندما تعمل العمليات على نفس النظام الطرفي، يتم الاتصال فيما بينها عن طريق الاتصال البيني للعمليات (interprocess communication)، باستخدام القواعد التي يحكمها نظام التشغيل على الوحدة الطرفية. لكننا في هذا الكتاب لسنا مهتمين بشكلٍ خاص بكيفية اتصال العمليات فيما بينها على نفس المضيف، وإنما بكيفية اتصال العمليات التي تعمل على مضيفين مختلفين (وقد تعمل تحت نظم تشغيل مختلفة).

تتصل العمليات على نظامين طرفيين مختلفين فيما بينها بتبادل الرسائل عبر شبكة الحاسب – فالعملية المُرسِلة تُنشئ وتُرسِل الرسائل إلى الشبكة، والعملية المُستقبِلة تستلم تلك الرسائل ومن المحتمل أن ترد عليها بإعادة رسائل للمُرسِل. كما هو موضح بالشكل 2-1 تتصل العمليات مع بعضها البعض ضمن طبقة التطبيقات في نموذج الخمس طبقات للبروتوكولات.

عمليتا الزبون والخادم

يتكون تطبيق الشبكة من أزواج من العمليات التي تتبادل الرسائل فيما بينها عبر الشبكة. على سبيل المثال في تطبيق الويب، تتبادل عملية المتصفّح الرسائل مع عملية خادم الويب، وفي نظام مشاركة النظائر للملفات يتم نقل الملف من عملية في نظيرٍ ما إلى عملية في نظيرٍ آخر. لكل زوج من العمليات المتصلة فيما بينها، تسمى إحدى العمليتين عادةً زبوناً والعملية الأخرى خادماً. فمثلاً في تطبيق الويب نعتبر المتصفّح عملية زبون وخادم الويب عملية خادم. وفي مشاركة النظائر للملفات، يُعتبَر النظير الذي يُنزِّل الملف (downloading) بمثابة الزبون، بينما يُعتبَر النظير الذي يُرسِل (يُحمِّل) الملف (uploading) بمثابة الخادم .

ولعلك لاحظت أنه في بعض التطبيقات، كمشاركة النظائر للملفات، يمكن أن تقوم عملية ما بدور الزبون والخادم في نفس الوقت. في الحقيقة يمكن أن تقوم عملية ما في نظام مشاركة الملفات بتحميل وتنزيل الملفات. ومع ذلك ففي سياق أي جلسة اتصال بين زوج من العمليات، يكون بوسعنا دائماً اعتبار إحدى العمليتين "زبون" والعملية الأخرى "خادم". نُعرِّف هنا عمليتي الزبون والخادم كالتالي:

> في جلسة اتصال بين زوج من العمليات، العملية التي تبدأ الاتصال (تتصل أولاً بالعملية الأخرى في بداية الجلسة) تُعَدّ الزبون، بينما العملية التي تنتظر لكي يُتصَل بها لتبدأ الجلسة تُعَدّ الخادم .

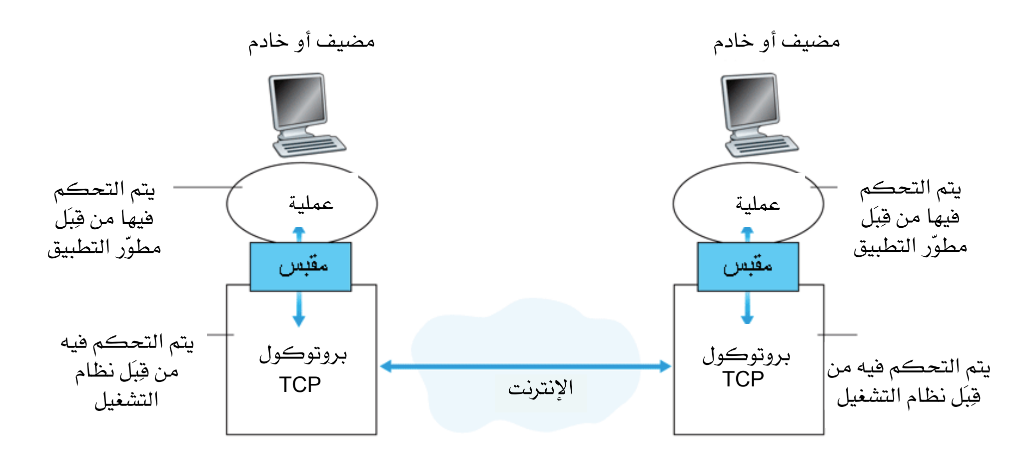
في الويب تبدأ عملية المتصفّح اتصالاً مع عملية خادم الويب؛ ولذا فعملية المتصفّح هي "الزبون" وعملية خادم الويب هي "الخادم". في مشاركة النظائر للملفات، عندما يسأل النظير A النظير B لإرسال ملف معين، فإنه في سياق جلسة الاتصال تلك يكون النظير A هو الزبون بينما النظير B هو الخادم. وعندما لا يكون هناك احتمال للبس أو عدم الوضوح في المعنى، سنستخدم أحياناً المصطلح "جانب الزبون" و"جانب الخادم" من التطبيق. في نهاية هذا الفصل سوف نستعرض خطوات برنامج بسيط لكلٍّ من جانبي الخادم والزبون لتطبيقات الشبكة.

الواجهة بين العملية وشبكة الحاسب

كما لاحظنا من قبل تتكون أكثر التطبيقات من أزواج من العمليات المتصلة فيما بينها، حيث تُرسِل كلٌّ من العمليتين الرسائل إلى العملية الأخرى. يجب أن تمر أي رسالة صادرة من عملية إلى أخرى عبر الشبكة التحتية. ترسل العملية الرسائل إلى الشبكة، وتتلقى الرسائل منها، خلال واجهة برمجة يطلق عليها مقبس (socket). دعنا نستعرض مثالاً يساعدنا على فهم العمليات والمقابس. تشبه العملية البيت، ومقبسها يمثل باب البيت. عندما تريد عملية إرسال رسالة إلى عملية أخرى تعمل على المضيف الآخر، تدفع العملية بالرسالة خارج بابها (مقبسها)، حيث تفترض العملية المُرسِلة هذه وجود بنية نقل تحتية على الجانب الآخر من بابها تقوم بنقل الرسالة إلى باب العملية المُرسَل إليها على مضيف الوجهة النهائية. وعندما تصل الرسالة إلى ذلك المضيف، فإنها تمر من خلال باب (مقبس) العملية المُستقبِلة، والتي تتصرف عندئذ بناءً على محتوى الرسالة.

يوضح الشكل 2-3 الاتصال بين مقبسي عمليتين تعملان على الإنترنت. يفترض الشكل أن نظام النقل التحتي المُستخدَم من قِبَل العمليتين هو بروتوكول التحكم في الإرسال (TCP)، وكما يوضح هذا الشكل، فإن مقبس الاتصال هو الواجهة بين طبقة التطبيقات وطبقة النقل على المضيف. كما يسمى المقبس أيضاً واجهة برمجة التطبيقات (API) بين التطبيق والشبكة، حيث إن المقبس هو واجهة البرمجة التي تُبنَى من خلالها تطبيقات الشبكة. وفي حين يمتلك مطوِّر التطبيقات

سيطرة كاملة على كل شيء على جانب طبقة التطبيق من المقبس، فإنه ليس لديه سوى القليل مما يمكنه عمله على جانب طبقة النقل من المقبس. تنحصر مجالات السيطرة الوحيدة التي يمتلكها مطوِّر التطبيقات على جانب طبقة النقل في: (1) اختيار نوع بروتوكول النقل، وربما (2) القدرة على تحديد قيمة بضعة متغيرات لطبقة النقل، كالسعة القصوى لذاكرة التخزين المؤقت (buffer) والحجم الأقصى لقطعة TCP (TCP segment) (سنتناولها في الفصل الثالث). وبمجرد اختيار مطوِّر التطبيقات بروتوكول النقل (في حالة توفر ذلك الخيار)، يُبنَى التطبيق باستعمال خدمات طبقة النقل المتوفرة مع ذلك البروتوكول. سنتناول المقابس بشيءٍ من التفصيل في الجزأين 2-7 و2-8.



الشكل 2-3 عمليتا التطبيق والمقابس وبروتوكول النقل بينهما.

2-1-3 خدمات النقل المتاحة للتطبيقات

تذكّر أن المقبس (socket) هو الواجهة بين التطبيق وبروتوكول طبقة النقل. على جانب الإرسال يدفع التطبيق بالرسائل خلال المقبس، وعلى الجهة الأخرى للمقبس يضطلع بروتوكول طبقة النقل بمسؤولية توصيل الرسائل إلى باب المقبس في جانب الاستقبال.

توفر العديد من الشبكات (بما فيها الإنترنت) أكثر من بروتوكول لطبقة النقل. وعندما تطور تطبيقاً ما يجب أن تختار أحد بروتوكولات طبقة النقل المتوفرة. كيف تقوم بهذا الاختيار؟ على الأغلب سوف تدرس الخدمات المقدمة من البروتوكولات المتوفرة في طبقة النقل، ثم تختار البروتوكول الذي يقدم خدمات أكثر ملاءمة لاحتياجات تطبيقك. هذا يشبه تماماً اختيارك وسيلة مواصلات (الطائرة أو القطار) للسفر بين مدينتين، حيث يجب أن تختار وسيلة دون الأخرى. كل وسيلة توفر خدمات مختلفة (على سبيل المثال محطات القطار كثيرة ويمكن أن تركب وتنزل من معظم أحياء المدينة، بينما تقطع الطائرة الرحلة في زمن أقصر).

ما هي الخدمات التي يمكن أن يقدمها بروتوكول طبقة النقل للتطبيقات التي تستخدمه؟ يمكن بشكلٍ عام تصنيف متطلبات الخدمة لتطبيق ما على أربعة محاور: النقل الموثوق للبيانات (reliable data transfer)، والطاقة الإنتاجية (throughput)، والتوقيت (timing)، والأمن (security).

النقل الموثوق للبيانات (Reliable Data Transfer)

كما تناولنا في الفصل الأول، من الممكن أن تفقد رزم البيانات في شبكة الحاسب. على سبيل المثال يمكن أن تُفقد البيانات نتيجة امتلاء ذاكرة المخزن المؤقت في موجّه، أو أن تُهمل عند الموجّه أو المضيف إثر اكتشاف خطأ في بعض بتاتها. وفي العديد من التطبيقات كالبريد الإلكتروني ونقل الملفات والوصول إلى المضيفات عن بُعد ونقل وثائق الويب والتطبيقات المالية، يمكن أن يؤدي فقد البيانات إلى نتائج وخيمة العاقبة (في الحالة الأخيرة إما للمصرف أو للعميل!). وهكذا لدعم تلك التطبيقات ينبغي اتخاذ الإجراءات اللازمة لضمان استقبال البيانات المُرسَلة من أحد طرفي التطبيق بشكلٍ صحيح وكامل على الطرف الآخر من التطبيق. إذا وفّر بروتوكول ما مثل تلك الخدمة للتوصيل المضمون للبيانات يُقال: إنه يوفِّر نقلاً موثوقاً للبيانات. إحدى الخدمات المهمة التي يمكن أن يوفرها

بروتوكول طبقة النقل لتطبيق ما هي النقل الموثوق للبيانات بين عملية وأخرى. وعندما يوفر بروتوكول نقل هذه الخدمة، لن يكون على العملية المُرسِلة سوى تمرير بياناتها إلى المقبس مع ثقة تامة في أن تلك البيانات ستصل كاملةً وبدون أخطاء إلى العملية المُستقبِلة.

عندما لا يوفر بروتوكول طبقة النقل نقلاً موثوقاً للبيانات من الممكن ألا تصل البيانات المُرسَلة إلى العملية المُستقبِلة على الإطلاق. قد يكون هذا الأمر مقبولاً لدى التطبيقات التي تتسامح بعض الشيء مع فقد البيانات، ومنها بشكلٍ خاص تطبيقات الوسائط المتعددة كالصوت والفيديو الفوري أو تسجيلات الصوت/الفيديو المُخزّنة والتي يمكن أن تسمح ببعض الفقد في البيانات. في تطبيقات الوسائط المتعددة تلك قد يؤدي فقد البيانات إلى خلل طفيف يمكن تقبله أثناء تشغيل تسجيلات الصوت/الفيديو – وهو أمرٌ لا يمثل عيباً كبيراً يضر بجودة البيانات عند استعادتها على الطرف الآخر.

الطاقة الإنتاجية (Throughput)

في الفصل الأول قدّمنا مفهوم الطاقة الإنتاجية المتاحة والتي – في سياق جلسة اتصال بين عمليتين على طول مسار في الشبكة – تمثل معدل توصيل البيانات إلى عملية المُستقبِل. ولما كانت الجلسات الأخرى تشارك في الحيّز الترددي (bandwidth) على طول المسار في الشبكة، ونظراً لأن تلك الجلسات تجيء وتذهب، يمكن أن تتفاوت الطاقة الإنتاجية المتوفرة من وقت لآخر. تقود هذه الملاحظات إلى خدمة طبيعية أخرى يمكن أن يوفرها بروتوكول طبقة النقل، ألا وهي ضمان توفير طاقة إنتاجية بمعدل محدد. بموجب هذه الخدمة يمكن أن يطلب التطبيق ضمان طاقة إنتاجية تعادل r بت/ثانية، وعندئذ يضمن بروتوكول طبقة النقل توفير طاقة إنتاجية لا تقل أبداً عن r بت/ثانية. إن مثل تلك الخدمة – أي توفير طاقة إنتاجية مضمونة – تروق للعديد من التطبيقات. على سبيل المثال إذا كوّد تطبيق هاتف الإنترنت الصوت بمعدل 32 كيلوبت/ثانية،

فإنه يحتاج لإرسال البيانات إلى الشبكة وتسليم البيانات إلى التطبيق المُستقبِل بنفس هذا المعدل. وإذا لم يستطع بروتوكول النقل توفير تلك الطاقة الإنتاجية فسيضطر التطبيق إما إلى تكويد البيانات بمعدل أقل (والحصول على طاقة إنتاجية كافية للمحافظة على معدل التكويد المنخفض هذا) أو التوقف تماماً، حيث إن تأمين نصف الطاقة الإنتاجية المطلوبة هو أمر قليل أو عديم الفائدة لتطبيق هاتف الإنترنت هذا. يُطلق على التطبيقات التي لها متطلبات تتعلق بالطاقة الإنتاجية تطبيقات حسّاسة للحيّز الترددي (bandwidth sensitive). يلاحظ أن العديد من التطبيقات الحالية للوسائط المتعددة هي من هذا النوع، رغم أن بعض تطبيقات الوسائط المتعددة قد تستعمل أساليب التكويد التكيفي، وذلك لتكويد البيانات بمعدل يجاري الطاقة الإنتاجية المتوفرة حالياً على الشبكة.

بينما تتميز التطبيقات الحسّاسة للحيّز الترددي بأن لها متطلبات محددة من الطاقة الإنتاجية، فإن التطبيقات المرنة (elastic applications) – على النقيض من ذلك – يمكن أن تستفيد من أي قدر من الطاقة الإنتاجية المتاحة قلّ ذلك أو كثر. وكمثال لذلك، فإن البريد الإلكتروني، ونقل الملفات، ونقل وثائق الويب، كلها تطبيقات من هذاالنوع المرن. بالطبع كلما زادت الطاقة الإنتاجية المتاحة كلما كان الوضع أفضل، وكما يقول المثل: لا يمكن أن تكون غنياً أكثر من اللازم، ولا نحيفاً أكثر من اللازم، ولا ذا طاقة إنتاجية أكثر من اللازم (أي لا يوجد حد أقصى للاكتفاء من هذه الأشياء)!

التوقيت (Timing)

يمكن أيضاً أن يوفر بروتوكول طبقة النقل ضمانات تتعلق بالتوقيت. وكما هو الحال مع ضمانات الطاقة الإنتاجية، فإن ضمانات التوقيت يمكن أن تأخذ العديد من الأشكال والصور. فمثلاً قد يشمل ذلك ضمان أن كل بت يرسلها المُرسِل إلى مقبس الإرسال تصل إلى مقبس المُستقبِل خلال زمن لا يتعدى 100 ميللي ثانية لاحقاً. تروق مثل تلك الخدمة للتطبيقات الفورية التفاعلية كهاتف الإنترنت والبيئات الافتراضية والمؤتمرات عن بُعد والألعاب متعددة اللاعبين،

وجميعها تضع قيوداً صارمة على توقيت تسليم رزم البيانات للحصول على تطبيق فعّال (راجع الفصل السابع، و[Gauthier 1999] و[Ramiee 1994]). على سبيل المثال تؤدي التأخيرات الطويلة في نقل بيانات هاتف الإنترنت إلى توليد وقفات غير طبيعية في المحادثة، كما أن التأخير الطويل في لعبة متعددة اللاعبين أو بيئة تفاعلية افتراضية بين فعل شيء ورؤية الاستجابة له من البيئة (على سبيل المثال من لاعب آخر في نهاية توصيلة من طرف إلى طرف) يجعل التطبيق يبدو أقل واقعية. في حالة التطبيقات غير الفورية، يُفضَّل التأخير الأقل دائماً على التأخير الأكبر، ولكن بدون فرض قيود صارمة على التأخيرات من طرف إلى طرف.

الأمن (Security)

أخيراً يمكن أن يزوِّد بروتوكول النقل تطبيقاً ما بواحدٍ أو أكثر من خدمات الأمن. على سبيل المثال في مضيف الإرسال يمكن أن يُشفّر بروتوكول النقل كل البيانات الصادرة من العملية المُرسِلة. وفي المقابل وعلى مضيف الاستقبال يقوم بروتوكول النقل باسترجاع البيانات الأصلية قبل تسليمها إلى العملية المستقبلة. توفر مثل هذه الخدمة سريّة للتعامل بين العمليتين، حتى لو كانت البيانات تجرى مراقبتها بطريقة ما بين العمليتين على المُرسِل والمُستقبِل. كما يمكن أن يوفر بروتوكول النقل خدمات أمن أخرى بالإضافة إلى السريّة كسلامة البيانات والتوثيق الطرفي، وسوف نتناول هذه الموضوعات بالتفصيل في الفصل الثامن.

2-1-4 خدمات النقل المتوفرة على الإنترنت

استعرضنا حتى الآن خدمات النقل التي يمكن أن توفرها شبكة الحاسب بصفة عامة. دعنا الآن نكون أكثر تحديداً بفحص أنواع الدعم المتاحة من شبكة الإنترنت للتطبيقات. توفر الإنترنت (وبعموم أكثر شبكات TCP/IP) بروتوكولي نقل لاستخدام التطبيقات، وهما: بروتوكول وحدة بيانات المُستخدِم (UDP) وبروتوكول التحكم في الإرسال (TCP). عندما تقوم (كمطوّر برامج)

بإنشاء تطبيق شبكة جديد للإنترنت، سيكون من أول القرارات التي يتعين عليك اتخاذها ما إذا كنت ستستخدم بروتوكول UDP أو TCP. يوفر كلٌّ من هذين البروتوكولين نموذج خدمة مختلف للتطبيقات التي تستخدمه. يوضح الشكل 2-4 متطلبات الخدمة لبعض التطبيقات المختارة.

التطبيق	فقد البيانات	الحيز الترددي	الحساسية للوقت
نقل الملفات	غير مسموح به	مرن	غير حساس
البريد الإلكتروني	غير مسموح به	مرن	غير حساس
الويب	غير مسموح به	مرن (بضعة كيلوبت/ثانية)	غير حساس
هاتف الإنترنت ومؤتمرات الفيديو	متساهل	الصوت: بضعة كيلوبت/ثانية – 1 ميجابت/ثانية الفيديو: 10 كيلوبت/ثانية – 5 ميجابت/ثانية	حساس (بضع مئات من الميللي ثانية)
ملفات الصوت والفيديو المخزنة	متساهل	الصوت: بضعة كيلوبت/ثانية – 1 ميجابت/ثانية الفيديو: 10 كيلوبت/ثانية – 5 ميجابت/ثانية	حساس (بضع ثوانٍ)
الألعاب التفاعلية	متساهل	بضعة كيلوبت/ثانية – 10 كيلوبت/ثانية	حساس (بضع مئات من الميللي ثانية)
المراسلة الفورية	غير مسموح به	مرن	تختلف فقد يكون بعضها حساس والبعض الآخر غير حساس

الشكل 2-4[1] متطلبات بعض تطبيقات الشبكة.

[1] يشار إلى بعض "الجداول" في الكتاب الأصلي بالأشكال، لذا تركنا الإشارة إليها "بالأشكال" من أجل عدم إحداث تغيير بتسلسل ترقيم الأشكال والجداول مما يسهل الرجوع للكتاب الأصلي (لمن أراد ذلك).

خدمات بروتوكول TCP

يتضمن نموذج خدمة بروتوكول TCP خدمة نقل توصيلية وخدمة نقل موثوق للبيانات. عندما يستدعي تطبيق ما بروتوكول النقل TCP فإنه يستفيد من كلٍّ من هاتين الخدمتين.

- خدمة النقل التوصيلية: في بروتوكول TCP يتبادل كلٌّ من الزبون والخادم معلومات التحكم قبل أن تبدأ رسائل التطبيق بالتدفق. من شأن هذا الإجراء الذي يعرف بالمصافحة (handshaking) بين الطرفين تنبيه كلٍّ من الزبون والخادم للاستعداد لتدفق رزم البيانات. عقب مرحلة المصافحة يقال إن توصيلة TCP قائمة بين مقابس العمليتين (لدى الزبون والخادم). وهذه التوصيلة من النوع المزدوج الاتجاه (full-duplex) مما يُمكِّن كلا العمليتين من إرسال الرسائل إلى بعضهما البعض في نفس الوقت وعلى نفس التوصيلة. عندما ينتهي التطبيق من إرسال الرسائل يجب عليه إغلاق التوصيلة. ويلاحظ أنه بالرغم من أننا أطلقنا على تلك الخدمة "خدمة توصيلية" إلا أننا نقصد في الواقع "خدمة مبنية على التوصيلة" حيث إن العمليتين موصلتان ببعضهما بطريقة فضفاضة للغاية. سنتناول في الفصل الثالث بالتفصيل الخدمة المبنية على التوصيل وكيفية تحقيقها.
- خدمة النقل الموثوق للبيانات: يمكن أن تعتمد العمليات المتصلة فيما بينها على بروتوكول التحكم في الإرسال (TCP) لتسليم كل البيانات التي أرسلت بدون خطأ أو تكرار وبالترتيب الصحيح. وعندما يرسل أحد طرفي التطبيق سلسلة من البايتات إلى مقبس اتصال، يمكن أن يعتمد على بروتوكول التحكم في الإرسال لتسليم نفس سلسلة البايتات إلى مقبس المُستقبِل بدون فقد أو تكرار فيها.

يتضمن بروتوكول TCP أيضاً آلية للتحكم في الازدحام (congestion) في الشبكة، والتي تأخذ بعين الاعتبار الصالح العام لمُستخدِمي الإنترنت ككل وليس فقط المنفعة المباشرة والقريبة للعمليتين المعنيتين فقط. تقوم آلية التحكم في

الازدحام بالحد من معدل إرسال البيانات من عملية الإرسال (على زبون أو خادم) عندما تكون الشبكة مزدحمة بين المُرسِل والمُستقبِل. وكما سنرى في الفصل الثالث، تحاول آلية التحكم في الازدحام في بروتوكول TCP تحقيق التوزيع العادل للحيّز الترددي المتاح بين توصيلات TCP. غير أن خَنْق معدل الإرسال يمكن أن يكون له تأثير ضار جداً على التطبيقات الفورية (تطبيقات الوقت الحقيقي (real-time applications)) كإرسال الصوت والفيديو، والتي تتطلب حداً أدنى من الحيّز الترددي. ولكن من ناحية أخرى، فإن التطبيقات الفورية متسامحة في فقد الرزم، ومن ثم فهي ليست بحاجة إلى خدمة نقل موثوقة تماماً. ولهذه الأسباب مجتمعة، فإن مطوّرى التطبيقات الفورية عادة ما يستخدمون بروتوكول UDP بدلاً من بروتوكول TCP.

خدمات بروتوكول UDP

يعتبر بروتوكول UDP بروتوكول نقل بسيط بأقل نموذج خدمة، فهو بروتوكول لاتوصيلي (connectionless)، حيث لا يتضمن إجراء مصافحة (handshaking) قبل بدأ الاتصال بين عمليتي التطبيق. يوفر هذا البروتوكول خدمة غير موثوقة لنقل البيانات. فعندما تُرسِل عملية ما رسالة إلى مقبس UDP، فإن البروتوكول لا يعطي أي ضمان لأن تصل الرسالة إلى عملية المُستقبِل. علاوة على ذلك قد تصل الرسائل إلى عملية المُستقبِل بترتيب مختلف.

لا يتضمن بروتوكول UDP أية آلية للتحكم في الازدحام في الشبكة، ولذا يمكن لعملية الإرسال بث البيانات إلى مقبس UDP بأي معدل تختاره (ولكن يجب ملاحظة أن الطاقة الإنتاجية المتحققة فعلياً من طرف إلى طرف قد تكون أقل من هذا المعدل بسبب الحيّز الترددي المحدود لوصلات الشبكة المُستخدَمة بين المُرسِل والمُستقبِل أو بسبب الازدحام). نظراً لأن التطبيقات الفورية غالباً ما تسمح ببعض الفقد في البيانات ولكنها تتطلب ضمان حد أدنى لمعدل الإرسال لتكون فعالة، يختار مطوّرو التطبيقات الفورية أحياناً تشغيل تطبيقاتهم على UDP، ومن ثم تفادي الأعباء الإضافية في TCP للتحكم في الازدحام وفي تركيبة الرزم. من

ناحية أخرى ونظراً لأن العديد من برامج الحماية (الجدران النارية أو الحواجز المنيعة) (firewalls) تُضبط لحجب أكثر أنواع حركة مرور بيانات UDP، لجأ المصممون على نحو متزايد مؤخراً لتشغيل تطبيقات الوسائط المتعددة والفورية على TCP [Sripanidkulchai 2004].

نبذة عن الأمن (Focus on Security)

تأمين TCP

لا يوفر أيٌ من البروتوكولين TCP أو UDP تشفيراً للبيانات (encryption)، أي أن البيانات التى ترسلها عملية الإرسال خلال المقبس هى نفسها البيانات التي تنقل خلال الشبكة لعملية الوجهة. لذلك فإنه على سبيل المثال إذا أرسلت عملية الإرسال كلمة سر غير مشفرة خلال المقبس فإنها ستُرسَل بنفس الشكل (أي غير مشفرة) خلال كل الوصلات في الشبكة بين المُرسِل والمُستقبِل. وبالتالي من المحتمل أن يتم التقاطها واكتشافها على أيٍ من تلك الوصلات البينية. ولأن الأمن والخصوصية أصبحت ذات أهمية عالية للعديد من التطبيقات طوّر مجتمع الإنترنت تحسيناً لبروتوكول TCP يسمى طبقة المقابس الآمنة (SSL). يقدم بروتوكول TCP المحسّن بطبقة المقابس الآمنة (SSL) كل شيء يقدمه بروتوكول TCP التقليدي. بالإضافة إلى توفير خدمات الأمن من عملية إلى عملية كتشفير البيانات (data encryption)، وسلامة البيانات (data integrity)، والتحقق من هوية الأطراف المتصلة (authentication). ونؤكد القول بأن SSL ليس بروتوكول نقل مثل TCP أو UDP وإنما هوتحسين لبروتوكول TCP، وهذا التحسين مطبق في طبقة التطبيقات. وبالتحديد إذا أراد تطبيق أن يستخدم خدمات SSL فإن عليه أن يستخدم تعليمات SSL (توجد مكتبات (libraries) وفصائل (classes) على درجة عالية من الأداء) في جانبي الزبون والخادم للتطبيق. لدى SSL واجهة برمجة API خاصة به مشابهة لواجهة برمجة المقابس التقليدية الخاصة ببروتوكول TCP. عندما يستخدم تطبيقٌ ما SSL فإن عملية الإرسال ترسل البيانات غير مشفرة إلى مقبس SSL والذي يقوم بدوره بتشفير البيانات وإرسالها إلى مقبس TCP. تمر البيانات المشفرة خلال شبكة الإنترنت حتى تصل إلى مقبس TCP لدى عملية الاستقبال فيرسلها مقبس TCP إلى SSL والذي يقوم باسترجاع البيانات الأصلية وتمريرها خلال مقبس SSL إلى عملية الاستقبال. سوف نغطي بعض تفاصيل SSL في الفصل الثامن.

الخدمات التي لا توفرها بروتوكولات النقل على الإنترنت

لقد رتبنا خدمات بروتوكول النقل الممكنة على أربعة محاور: النقل الموثوق للبيانات، والطاقة الإنتاجية، والتوقيت، والأمن. أي من تلك الخدمات يوفرها بروتوكول TCP وأي منها يوفرها بروتوكول UDP؟ لاحظنا أن بروتوكول TCP يتيح نقلاً موثوقاً للبيانات من طرف إلى طرف. كما أنه يمكن بسهولة استخدام SSL لتحسين أداء بروتوكول TCP من منظور طبقة البرامج وذلك بتوفير خدمات لأمن البيانات. غير أنه في وصفنا المقتضب لبروتوكول TCP وبروتوكول UDP غاب بشكلٍ واضح أي ذِكر لخدمات ضمان الطاقة الإنتاجية أو التوقيت، حيث أن تلك الخدمات غير متوفرة في أي من بروتوكولات النقل الخاصة بالإنترنت اليوم. هل يعني ذلك أنه ليس بإمكان التطبيقات الحسّاسة للتوقيت كهاتف الإنترنت أن تعمل على إنترنت اليوم؟ واضح أن الجواب لا، فالإنترنت تستضيف تطبيقات حسّاسة للتوقيت لسنوات عديدة خلت. تعمل تلك التطبيقات بشكلٍ لا بأس به في أغلب الأحيان لأنها صُمِّمت لتحمُّل النقص في الضمان الذي تتطلبه إلى أقصى حد ممكن. سوف نناقش العديد من حيل التصميم المُستخدَمَة في هذا المجال في الفصل السابع. ومع ذلك فالتصميم الماهر له حدوده عند الزيادة المفرطة في زمن التأخير كما هو الحال عادة في الإنترنت العامة. وخلاصة القول هي أن إنترنت اليوم يمكن أن تقدّم خدمة مُرضية في أغلب الأحيان للتطبيقات الحسّاسة للتوقيت لكنها لا تستطيع تقديم أي ضمانات فيما يتعلق بالحيّز الترددي أو التوقيت.

يبين الشكل 2-5 بروتوكولات النقل المستعملة من قِبَل بعض تطبيقات الإنترنت الشائعة. نلاحظ أن تطبيقات البريد الإلكتروني والوصول إلى الحاسبات عن بُعد والويب ونقل الملفات تستخدم بروتوكول TCP. لقد اختارت تلك التطبيقات TCP أساساً لأنه يوفر خدمة نقل موثوق للبيانات تضمن وصول كل البيانات إلى وجهتها النهائية في نهاية الأمر. ونلاحظ أيضاً أن هاتف الإنترنت يعمل عادة على بروتوكول UDP. يحتاج كل جانب من تطبيق هاتف الإنترنت لبث

البيانات عبر الشبكة بحدٍّ أدنى يجب توفره لمعدل الإرسال (راجع التسجيل الصوتي الفوري في الشكل 2-4)، واحتمال تحقق ذلك مع UDP أرجح منه مع TCP. كما أن تطبيقات هاتف الإنترنت متسامحة في فقد بعض البيانات، ومن ثم فهي ليست بحاجة ماسّة إلى خدمة النقل الموثوق للبيانات التي يوفرها بروتوكول TCP.

التطبيق	بروتوكول طبقة التطبيقات	بروتوكول طبقة النقل
البريد الإلكتروني	(RFC 2821) SMTP	TCP
الوصول إلى الحاسبات عن بُعد	(RFC 854) Telnet	TCP
الويب	(RFC 2616) HTTP	TCP
نقل الملفات	(RFC 959) FTP	TCP
تطبيقات الوسائط المتعددة	HTTP (كـ YouTube)، RTP	TCP أو UDP
هاتف الإنترنت	SIP، RTP، أو ذات ملكية خاصة كـ Skype	في العادة UDP

الشكل 2-5 تطبيقات الإنترنت الشائعة وبروتوكولات طبقتي التطبيقات والنقل التي تستخدمها.

عنونة العمليات (Addressing processes)

ركزت مناقشتنا حتى الآن على خدمات نقل البيانات بين عمليتين تتصلان فيما بينهما. لكن كيف تقوم عملية ما بتحديد العملية الأخرى التي تريد الاتصال معها بواسطة تلك الخدمات؟ كيف تحدد عملية تعمل على مضيف في مدينة أمهرست بولاية ماسوشوستس في الولايات المتحدة الأمريكية بأنها تريد الاتصال مع عملية معينة تعمل على مضيف في مدينة بانكوك بتايلند؟ لتعيين العملية التي ستتسلم البيانات، نحتاج إلى معلومتين: (1) اسم أو عنوان المضيف، (2) رقم معرّف يحدد العملية المُستلِمة على مضيف الوجهة.

يُميَّز المضيف في الإنترنت بعنوان IP. سنناقش عناوين IP بمزيد من التفصيل في الفصل الرابع. أما الآن فكل ما نحتاج لمعرفته هو أن عنوان IP هو رقم يتألف من 32 بتاً، ويمكن اعتباره كطريقة لتمييز المضيف عن غيره من عقد الشبكة تمييزاً فريداً. (ومع ذلك، وكما سنرى في الفصل الرابع، فمع استخدام مترجمات عناوين الشبكة (NATs) على نطاق واسع، أصبح عنوان IP المؤلف من 32 بتاً لا يكفي وحده من الناحية العملية لعنونة المضيف بطريقة فريدة).

بالإضافة إلى معرفة عنوان مضيف الوجهة النهائية، على المضيف المُرسِل أيضاً أن يُميِّز العملية المُستلِمة التي تعمل على مضيف الوجهة. هذه المعلومات مطلوبة لأنه بصفة عامة يمكن أن يُشغِّل مضيف الوجهة العديد من تطبيقات الشبكة في نفس الوقت. يؤدي هذا الغرض رقم منفذ الوجهة، وقد خُصّصت أرقام منافذ معينة للتطبيقات الشائعة. على سبيل المثال يُميَّز خادم الويب برقم المنفذ 80، بينما تُميَّزعملية خادم البريد (والتي تستخدم بروتوكول SMTP) برقم المنفذ 25. يمكنك الاطلاع على قائمة بأرقام المنافذ المشهورة لكل بروتوكولات الإنترنت المعيارية على الموقع http://www.iana.org. عندما يُنشئ مطوّر تطبيقات تطبيق شبكة جديد يجب أن يُخصّص له رقم منفذ جديد. وسوف نتناول أرقام المنافذ بالتفصيل في الفصل الثالث.

5-1-2 بروتوكولات طبقة التطبيقات

عرفنا قبل قليل أن عمليات الشبكة تتصل فيما بينها بإرسال الرسائل إلى المقابس. لكن كيف تصاغ هذه الرسائل، وما معاني الحقول المختلفة فيها؟ ومتى ترسلها العمليات؟ هذه الأسئلة تقودنا إلى عالم بروتوكولات طبقة التطبيقات. يحدد بروتوكول طبقة التطبيقات كيف ترسل عمليات التطبيق – والتي تعمل على الأنظمة الطرفية المختلفة – رسائل إلى بعضها البعض. وبالتحديد فإن بروتوكول طبقة التطبيقات يُعرِّف ما يلي:

- أنواع الرسائل المتبادلة، كرسائل الطلب ورسائل الرد.
- صيغ الرسائل المختلفة، كالحقول الموجودة بالرسالة والفواصل بينها.

- معاني الحقول، أي معنى المعلومة في كل حقل من حقول الرسالة.
- القواعد التي تحكم متى وكيف تُرسِل عملية ما الرسائل وترد عليها.

يتم توصيف بعض بروتوكولات طبقة التطبيقات في وثائق RFCs (طلبات تعليقات) متاحة للجمهور. على سبيل المثال يوجد بروتوكول طبقة التطبيقات الخاص بالويب HTTP (بروتوكول نقل مادة الإنترنت) في طلب التعليقات RFC 2616. وبالتالي فإذا ما اتبع مطوّر لمتصفّح جديد للإنترنت قواعد HTTP المذكورة في تلك الوثيقة، فإن المتصفّح الجديد سيتمكن من استجلاب صفحات الويب من أي خادم ويب يتبع نفس قواعد HTTP المذكورة في تلك الوثيقة. يلاحظ أن العديد من البروتوكولات الأخرى لطبقة التطبيقات تعتبر ذات ملكية خاصة للشركات (proprietary)، وغير متاحة للجمهور عن قصد. فعلى سبيل المثال يستخدم العديد من أنظمة مشاركة النظائر للملفات بروتوكولات خاصة لطبقة التطبيقات.

من المهم التمييز بين تطبيقات الشبكة وبروتوكولات طبقة التطبيقات، فبروتوكول طبقة التطبيقات يمثل فقط أحد أجزاء تطبيق الشبكة (رغم كونه جزءاً هاماً بلا شك). دعنا نستعرض مثالين:

- الأول: الويب هو تطبيق زبون/خادم يسمح للمُستخدِم بالحصول على الوثائق من خادمات الويب عند الطلب، ويتكون من العديد من المكونات التي تضم: معياراً لصياغة وثيقة الويب (مثل HTML)، ومتصفّحات الويب (كمتصفّح نيتسكاب (Netscape Navigator) ومتصفّح مايكروسوفت (Microsoft Internet Explorer))، وخادمات الويب (مثل Apache، مايكروسوفت، وخادمات نيتسكاب)، وبروتوكول طبقة التطبيقات (HTTP). يحدد بروتوكول طبقة التطبيقات للويب HTTP صيغة وتسلسل الرسائل المتبادلة بين متصفّح وخادم الويب. وهكذا فإن HTTP يمثل جزءاً واحداً من أجزاء تطبيق الويب.
- الثاني: تطبيق البريد الإلكتروني يتكون أيضاً من العديد من المكوّنات، تضم خادمات البريد التي تحوي صناديق بريد الوارد والصادر للمُستخدِمين، وقارئ البريد الذي يسمح للمُستخدِم بقراءة وإنشاء الرسائل،

ومعياراً لتحديد هيكل وصيغة رسائل البريد الإلكتروني، وبروتوكولات طبقة التطبيقات والتي تحدد كيفية تبادل الرسائل بين الخادمات، وكيفية تبادل الرسائل بين الخادمات وبرامج قراءة البريد، وكيفية تفسير محتويات أجزاء معينة من رسالة البريد (على سبيل المثال سطور الترويسة للرسالة (header lines)). إن بروتوكول طبقة التطبيقات الرئيسي للبريد الإلكتروني هو SMTP (بروتوكول نقل البريد البسيط) [RFC 2821]. وهكذا فإن بروتوكول طبقة التطبيقات الرئيس للبريد الإلكتروني SMTP هو مجرّد جزء واحد (رغم كونه جزءاً هاماً) من أجزاء تطبيق البريد الإلكتروني.

6-1-2 تطبيقات الشبكة التي سنتناولها في هذا الكتاب

يتم تطوير تطبيقات جديدة للإنترنت كل يوم، بعضها عام وبعضها خاص لشركات معينة. وبدلاً من تغطية عدد كبير من تطبيقات الإنترنت بأسلوب موجز، فقد اخترنا أن نركز على عددٍ محدود من التطبيقات الهامة واسعة الانتشار. سنناقش في هذا الفصل خمسة تطبيقات مهمة: الويب، ونقل الملفات، والبريد الإلكتروني، وخدمة الدليل للإنترنت، ومشاركة النظائر للملفات. سنبدأ بمناقشة الويب، ليس فقط لأنه تطبيق منتشر على نطاق واسع، ولكن أيضاً لأن بروتوكوله (HTTP) بسيط وسهل الفهم. بعد ذلك سنتناول بروتوكول FTP بإيجاز، لأنه يعطينا مقارنة تباينية لطيفة مع HTTP. ثم نناقش البريد الإلكتروني – أول التطبيقات الحيوية للإنترنت – وهو أكثر تعقيداً من الويب، حيث إنه لا يستخدم بروتوكولاً واحداً فقط، وإنما يستخدم عدة بروتوكولات في طبقة التطبيقات. بعد البريد الإلكتروني سنتناول بروتوكول DNS والذي يوفر خدمة دليل الإنترنت لأسماء النطاقات. لا يتعامل معظم مُستخدِمي الإنترنت مباشرة مع الـ DNS، وإنما يستخدمون الـ DNS بشكلٍ غير مباشر من خلال التطبيقات الأخرى (كالويب، ونقل الملفات، والبريد الإلكتروني). ويوضح الـ DNS بشكلٍ رائع كيف أن جزءاً من الوظائف الرئيسية لقلب الشبكة (الترجمة من "اسم على

الشبكة" إلى "عنوان على الشبكة") يمكن أن يتم في طبقة التطبيقات على الإنترنت. وأخيراً سنتناول مشاركة النظائر للملفات، والتي تمثل الفئة الأكثر انتشاراً من تطبيقات الإنترنت اليوم حسب بعض القياسات (كقياس حركة مرور البيانات على الشبكة).

2-2 شبكة الويب وبروتوكول HTTP

حتى أوائل التسعينيات كانت الإنترنت تُستخدم أساساً من قِبَل الباحثين والأكاديميين وطلاب الجامعات للوصول للحاسبات عن بُعد، ولنقل الملفات من المضيفات المحلية إلى المضيفات البعيدة والعكس بالعكس، ولاستقبال وإرسال الأخبار والبريد الإلكتروني. ورغم أن هذه التطبيقات كانت (وستظل) مفيدة جداً، إلا أن الإنترنت كانت في واقع الأمر أيامها مجهولة خارج نطاق المجتمع الأكاديمي والبحثي. بعد ذلك ظهرت الشبكة العنكبوتية العالمية (الويب) [Berners-Lee 1994] إلى حيّز الوجود في أوائل التسعينيات كتطبيق جديد وهام للإنترنت والذي شد انتباه جمهور الناس إلى الإنترنت. لقد غيّر الويب بشكلٍ ملحوظ كيفية تفاعل الناس داخل وخارج بيئات عملهم. لقد نقل هذا التطبيق الإنترنت بشكلٍ أساسي من كونها مجرد إحدى شبكات البيانات العديدة إلى كونها شبكة البيانات الواحدة والفريدة.

لعل ما يروق لأكثر مُستخدِمي الويب هو كونه يعمل حسب الطلب، فهم يستقبلون ما يريدونه عندما يريدونه، بخلاف الإذاعة والتلفزيون حيث يُجبرون على توليف أجهزتهم لاستقبال البرامج عندما يقوم موفّرو المحتوى ببث تلك البرامج وإتاحة محتواها للجمهور. وبالإضافة إلى توفر المحتوى حسب الطلب، فللويب العديد من الميزات الرائعة الأخرى التي يحبها الناس ويقدرونها. فمن السهل جداً لأي فرد إضافة معلومات وجعلها متاحة عبر تلك الشبكة العنكبوتية، فبوسع كل شخص أن يصبح ناشراً بتكلفة زهيدة للغاية. فالوصلات التشعبية (hyperlinks) ومحركات البحث (search engines) تساعدنا على أن نبحر خلال محيط من مواقع الويب. والرسومات تحرك حواسنا، والأشكال وبرامج جافا

التفاعلية والعديد من الأدوات الأخرى تمكننا من التفاعل مع الصفحات والمواقع، كما يوفر الويب واجهة قوائم (menu interface) لِكَم هائل من المواد المسموعة والمرئية (audio and video) المخزنة على الإنترنت بصيغة الوسائط المتعددة (multimedia) والتي يمكن الوصول إليها حسب الطلب.

2-2-1 نظرة عامة على بروتوكول HTTP

يعتبر بروتوكول نقل مادة الإنترنت HTTP – بروتوكول طبقة التطبيقات للويب – بمثابة القلب من الشبكة العنكبوتية (الويب). ويوصّف هذا البروتوكول في RFC 1945 وRFC 2616، وينفّذ في برنامجين: برنامج الزبون وبرنامج الخادم. يتصل برنامجا الزبون والخادم (واللذان يعملان على نظامين طرفيين مختلفين) فيما بينهما بتبادل رسائل HTTP. ويتضمن بروتوكول HTTP تحديد الهيكل البنائي (النسق) لتلك الرسائل، وكيفية تبادلها بين برنامجي الزبون والخادم. قبل توضيح HTTP بالتفصيل ينبغي أن نراجع بعض مصطلحات الويب.

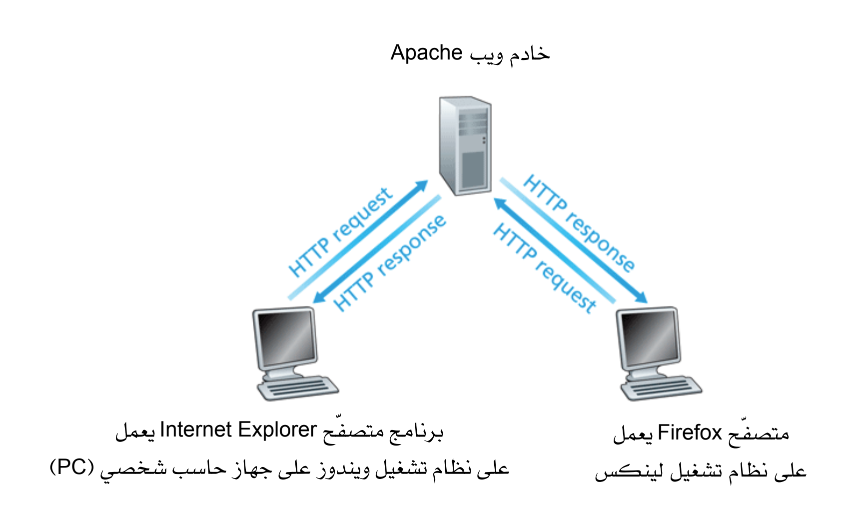
تتكون صفحة الويب – والتي تسمى أيضاً وثيقة (document) – من عناصر يطلق عليها كائنات (objects)، والكائن ببساطة هو ملف له عنوان URL وحيد، كملف HTML أو ملف صورة JPEG أو ملف برنامج جافا أو ملف مقطع فيديو (video clip). تشمل معظم صفحات الويب ملف HTML أساسي ومراجع (عناوين) لعدة كائنات. على سبيل المثال إذا كانت صفحة ويب تتضمن نص HTML وخمس صور JPEG فإنها تتكون من ستة كائنات: ملف HTML الأساسي بالإضافة إلى ملفات للصور الخمس. يرتبط ملف HTML الأساسي بالكائنات الأخرى في الصفحة عن طريق عناوين الـ URL للكائنات. يتألف عنوان URL من مكونين: اسم المضيف للخادم الذي يأوي الكائن، واسم المسارالخاص بالكائن (أي مكان تواجده على المضيف). على سبيل المثال يتكون العنوان:

http://www.someSchool.edu/someDepartment/picture.gif

من www.someSchool.edu والذي يمثل اسم المضيف للخادم، و/someDepartment/picture.gif يمثل اسم المسار. نظراً لأن متصفّحات الويب (مثل

Explorer وFirefox) تطبق جانب الزبون لبروتوكول HTTP، فسنستخدم (في سياق الويب) مصطلحي المتصفّح (browser) والزبون بالتبادل للدلالة على نفس الشيء. أما خادمات الويب (Web servers) — والتي تطبق جانب الخادم لبروتوكول HTTP — فتأوي كائنات الويب، حيث يوجد لكلٍ منها عنوان URL. من خادمات الويب الشائعة: Apache، وMicrosoft Internet Information Server (IIS).

يحدد بروتوكول HTTP كيفية طلب زبائن الويب صفحات الويب من خادمات الويب، وسوف نناقش التفاعل بين الزبون والخادم بالتفصيل لاحقاً، لكن الفكرة العامة موضحة في الشكل 2-6. فعندما يطلب مُستخدِم الإنترنت صفحة ويب (على سبيل المثال بالضغط على وصلة تشعبية (رابط) (hyperlink))، يرسل المتصفّح إلى الخادم رسائل طلب HTTP للحصول على الكائنات الموجودة في الصفحة المطلوبة. يستقبل الخادم بدوره تلك الطلبات ويستجيب برسائل رد HTTP تتضمن الكائنات المطلوبة.



الشكل 2-6 رسائل الطلب والرد لبروتوكول HTTP.

يعتمد HTTP على TCP كبروتوكول نقل أساسي (بدلاً من تشغيله فوق بروتوكول UDP). يبدأ زبون HTTP أولاً بالاتصال بالخادم عن طريق بروتوكول TCP، وعندما يتحقق الاتصال فإن برنامجيْ المتصفّح والخادم يستخدمان TCP من خلال واجهات المقبس. وكما وصفنا في الجزء 2-1، تُعتبر واجهة المقبس على جانب الزبون بمثابة المدخل بين عملية الزبون وتوصيلة TCP؛ وعلى جانب الخادم تعتبر واجهة المقبس المدخل بين عملية الخادم وتوصيلة TCP. يرسل الزبون رسائل طلب HTTP إلى واجهة مقبسه ويتلقى رسائل رد HTTP من خلال نفس الواجهة. وبنفس الطريقة يستقبل خادم HTTP رسائل طلب عبر واجهة مقبسه ويرسل رسائل الرد من خلالها أيضاً. وبمجرد أن يبعث الزبون رسالة إلى واجهة مقبسه فإن الرسالة تصبح خارج يده وتكون تحت سيطرة بروتوكول TCP. تذكّر من الجزء 1-2 أن بروتوكول TCP يوفر خدمة نقل موثوق للبيانات لبروتوكول HTTP. وهذا يعني ضمناً أن كل رسالة طلب HTTP أُرسِلت من عملية الزبون ستصل سليمة في النهاية إلى الخادم. وبنفس الطريقة فإن كل رسالة رد HTTP يرسلها الخادم ستصل للزبون سليمة في النهاية. تتضح هنا إحدى المزايا الهامة للبنية الطبقية لبروتوكولات الشبكة؛ فلا يلزم بروتوكول HTTP أن يكترث للبيانات التي قد تفقد، ولا بتفاصيل كيفية استعادة TCP لها، ولا بإعادة ترتيب البيانات ضمن الشبكة، فتلك مهمة بروتوكول TCP والبروتوكولات من أسفله في الطبقات الدنيا لرصة البروتوكولات (protocol stack).

من المهم ملاحظة أن الخادم يرسل الملفات المطلوبة إلى الزبائن بدون تخزين لأية معلومات عن الحالة (state information) فيما يتعلق بطلبات الزبائن، فمثلاً إذا طلب زبون ما نفس الكائن مرتين في غضون بضع ثوانٍ، فلن يرد الخادم بأنه قد سبق وأرسل ذلك الكائن إلى الزبون منذ قليل، بل سيرسل الخادم الكائن مرة أخرى لأنه قد نسي تماماً ما قام بعمله في السابق. ونظراً لأن خادم HTTP لا يحتفظ بأية معلومات عن الزبائن لذا يوصف بأنه بروتوكول بدون حالة (stateless). كما يجدر بنا الإشارة إلى أن الويب يستخدم بنية زبون/خادم كما

تقدّم وصفه في الجزء 2-1. كما أن خادم الويب يعمل دائماً وله عنوان IP ثابت، ويقوم على خدمة ملايين الطلبات المحتملة من المتصفّحات المختلفة للزبائن.

2-2-2 التوصيلات الدائمة والتوصيلات غير الدائمة (Non-Persistent and Persistent Connections)

يتواصل الزبون والخادم في العديد من تطبيقات الإنترنت لفترات طويلة بإرسال الزبون سلسلة من الطلبات واستجابة الخادم لكلٍ منها. وعلى حسب التطبيق وكيفية استخدامه فإن سلسلة الطلبات قد ترسل واحداً تلو الآخر (back-to-back)، أو بشكلٍ دوري على فترات منتظمة، أو بشكلٍ متقطع. وعند حدوث هذا التفاعل بين الخادم والزبون على بروتوكول TCP يتعين على مطوّري التطبيق اتخاذ قرار هام ألا وهو هل ينبغي إرسال كل زوج من رسائل الطلب والرد باستخدام توصيلة TCP منفصلة أو إرسال كل الطلبات وردودها على نفس توصيلة TCP؟ في الطريقة الأولى يقال إن التطبيق يستخدم توصيلات غير دائمة (non-persistent connections)؛ وفي الطريقة الثانية يقال إنه يستخدم توصيلات دائمة (persistent connections). ولفهم هذه القضية بعمق، دعنا نستعرض ميزات وعيوب الاتصالات الدائمة في سياق تطبيق HTTP، والذي يمكن أن يستخدم كلتا الطريقتين (الدائمة وغير الدائمة) للاتصال. فرغم أن بروتوكول HTTP يستخدم التوصيلات الدائمة في النمط الاعتيادي (default mode)، فإنه من الممكن تغيير تهيئة الخادم والزبون لاستخدام التوصيلات غير الدائمة بدلاً من ذلك.

بروتوكول HTTP بتوصيلات غير دائمة

دعنا نتتبع خطوات نقل صفحة ويب من الخادم إلى المتصفّح في حالة التوصيلات غير الدائمة. افترض أن الصفحة تتكون من ملف HTML أساسي وعشر صور JPEG، وأن كلاً من تلك الكائنات الأحد عشر موجودة على نفس الخادم، ولو فرضنا أن عنوان URL للملف HTML الأساسي هو:

http://www.someSchool.edu/someDepartment/home.index

فإن الخطوات تتم كالتالي:

1. تُنشئ عملية زبون HTTP توصيلة TCP مع الخادم www.someSchool.edu من خلال المنفذ رقم 80 (منفذ HTTP المعتاد) وكنتيجة لذلك سيصبح هناك مقبس اتصال (socket) بالزبون ومقبس اتصال بالخادم.
2. يُرسل زبون HTTP رسالة طلب HTTP إلى الخادم عن طريق مقبس الاتصال لديه تتضمن مسار الملف /someDepartment/home.index (سنتناول بعض تفاصيل رسائل HTTP لاحقاً).
3. يستقبل خادم HTTP رسالة الطلب عن طريق المقبس لديه ويحصل على الملف /someDepartment/home.index من وحدة التخزين لديه (ذاكرة RAM أو القرص الصلب)، ثم يُضمِّنه في رسالة رد HTTP، ويرسلها إلى الزبون عن طريق مقبس الاتصال لديه.
4. يطلب خادم HTTP من بروتوكول TCP إغلاق توصيلة TCP. (ولكن بروتوكول TCP لا يغلق التوصيلة في واقع الأمر حتى يتأكد من أن الزبون تسلم رسالة الرد سليمة).
5. يستقبل زبون HTTP رسالة الرد وبعدها ينهي بروتوكول TCP الاتصال. وتشير رسالة الرد إلى أن الكائن عبارة عن ملف HTML، فيقوم الزبون باستخراج الملف من رسالة الرد ثم يفحص محتواه ليجد عناوين صور JPEG العشرة.
6. تتكرر الخطوات الأربع الأولى لكل عنوان من عناوين صور JPEG.

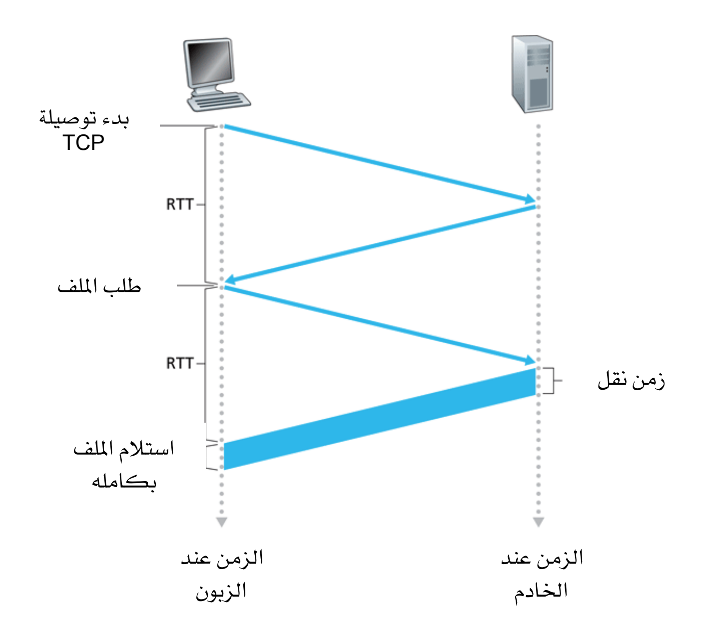
عندما يستقبل المتصفّح صفحة الويب يقوم بعرضها للمُستخدِم. ويمكن أن تختلف ترجمة متصفّحين مختلفين لصفحة ويب واحدة (أي يمكن أن يعرضاها للمُستخدِم بطرق مختلفة بعض الشيء). ليس لـ HTTP أية علاقة بكيفية تَرجمة صفحة الويب من قِبَل الزبون، فمواصفات HTTP المنشورة في RFC 1945 و RFC 2616 تُعرِّف فقط بروتوكول الاتصال بين برنامج زبون HTTP وبرنامج خادم HTTP.

توضح الخطوات المذكورة أعلاه استخدام التوصيلات غير الدائمة (non-persistent connections)، حيث إن كل توصيلة TCP تُغلَق بعد أن يرسل الخادم الكائن ولا تستمر التوصيلة لنقل الكائنات الأخرى. لاحظ أن كل توصيلة TCP تنقل فقط رسالة طلب واحدة ورسالة رد واحدة. وهكذا فعندما يطلب المُستخدِم صفحة الويب في المثال السابق فإنه يتم إنشاء إحدى عشرة توصيلة TCP.

في الخطوات المذكورة أعلاه، تعمّدنا عدم توضيح ما إذا كان الزبون يحصل على العشر صور باستخدام عشر توصيلات TCP متسلسلة، أو يحصل على البعض منها على توصيلات TCP متوازية. في الحقيقة يمكن أن يهيئ المُستخدِمون المتصفّحات الحديثة للتحكم في درجة التوازي. وفي أنماط الاستخدام المعتادة تفتح معظم المتصفّحات من خمس إلى عشر توصيلات TCP على التوازي، يقوم كلٌّ منها بالتعامل مع طلب واحد والرد عليه، وإذا أراد المُستخدِم يمكن ضبط العدد الأقصى لتوصيلات TCP المتوازية إلى واحد، وفي هذه الحالة يتم إنشاء التوصيلات العشر بشكلٍ متسلسل. وكما سنرى في الفصل القادم يُقصِّر استعمال التوصيلات المتوازية من زمن الاستجابة.

قبل الاستمرار في الشرح دعنا نقدر بشكلٍ تقريبي الوقت الذي ينقضي منذ أن يطلب الزبون ملف HTML الأساسي إلى أن يستقبله بالكامل. للقيام بذلك سنُعرِّف أولاً وقت الرحلة ذهاباً وإياباً ((Round-Trip-Time (RTT)) على أنه الوقت المُستغرَق لنقل رزمة بيانات صغيرة من الزبون إلى الخادم ثم عودتها بعد ذلك إلى الزبون. يشمل هذا الوقت زمن انتقال الرزمة (propagation delay)، والزمن الذي تقضيه الرزمة في صفوف الانتظار (queuing delay) عند الموجّهات والمحوّلات (routers and switches)، وزمن معالجة الرزمة (processing delay)، وقد سبق مناقشة تلك الأزمنة في الجزء 1-4. لنر ما يحدث عندما ينقر المُستخدِم على وصلة تشعبية. كما يوضح الشكل 2-7 يُنشئ المتصفّح توصيلة TCP بينه وبين خادم الويب، ويتضمن ذلك "مصافحة ثلاثية"؛ حيث يرسل الزبون قطعة بيانات TCP صغيرة إلى الخادم فيقر الخادم باستلام الرسالة، ويرد بقطعة بيانات TCP صغيرة، وأخيراً يقر الزبون باستلامه القطعة من الخادم. يستغرق أول جزأين من المصافحة

الثلاثية زمن RTT واحد. ثم يرسل الزبون رسالة طلب HTTP مدمجة مع الجزء الثالث من المصافحة الثلاثية (إشعار الاستلام) عبر توصيلة TCP. فور وصول رسالة الطلب إلى الخادم، يرسل الخادم ملف HTML إلى توصيلة TCP، حيث يستغرق ذلك زمن RTT آخر. وهكذا فإن الزمن الكلي يعادل ضعف RTT تقريباً، بالإضافة إلى زمن إرسال ملف HTML عند الخادم.



الشكل 2-7 حساب الزمن المستغرق لطلب ملف HTML والحصول عليه.

بروتوكول HTTP بتوصيلات دائمة

إن للتوصيلات غير الدائمة بعض العيوب. أولاً: يجب إنشاء توصيلة جديدة والإبقاء عليها لكل كائن مطلوب. تتطلب كلٌّ من تلك التوصيلات تخصيص مساحة من الذاكرة للتخزين المؤقت والاحتفاظ بمتغيرات TCP في كلٍّ من الزبون والخادم. يمكن أن يُشكِّل ذلك عبئاً كبيراً على خادم الشبكة، والذي قد يقوم

على خدمة طلبات من مئات الزبائن المختلفين في نفس الوقت. ثانياً: يواجه كل كائن – كما وضحنا آنفاً – زمن تأخير للتوصيل يعادل ضعف RTT (واحد لإنشاء توصيلة TCP وواحد لطلب واستلام الكائن).

أما في حالة التوصيلات الدائمة، فيترك الخادم توصيلة TCP مفتوحة بعد إرسال الرد وذلك لإرسال الطلبات والردود اللاحقة بين نفس الزبون والخادم. وبالتحديد، يمكن إرسال صفحة ويب كاملة (في المثال السابق ملف HTML الأساسي والصور العشرة) على توصيلة TCP دائمة واحدة. وعلاوة على ذلك يمكن إرسال عدد من صفحات ويب الموجودة على نفس الخادم من الخادم إلى نفس الزبون عبر توصيلة TCP دائمة واحدة. ويمكن إرسال الطلبات الخاصة بتلك الكائنات متلاصقة الواحد تلو الآخر، بدون انتظار للرد على الطلبات الجاري تنفيذها بطريقة خط الأنابيب (pipelining). عادةً ما يغلق خادم HTTP التوصيلة عند بقائها غير مستعملة لوقت معين (مدة انقضاء فترة الموقت ويمكن تغييرها). وعندما يستلم الخادم طلبات الكائنات متلاصقة فإنه يُرسل أيضاً الكائنات متلاصقة الواحد تلو الآخر. يستخدم بروتوكول HTTP عادة توصيلات دائمة بطريقة خط الأنابيب. وسوف نقارن أداء التوصيلات غير الدائمة والتوصيلات الدائمة بشكلٍ كمّي في تمارين الفصل الثانى والثالث. كما نشجعك على مراجعة [Heidemann 1997; Nielsen 1997].

2-2-3 صيغة رسائل HTTP

تتضمن مواصفات HTTP (في RFC 2616) تحديداً لصيغ رسائل HTTP، وهي على نوعين: رسائل طلب ورسائل رد، وسيتم مناقشتهما فيما يلى.

رسائل طلب HTTP

يوضح المثال التالي نموذجاً نمطياً لرسالة طلب HTTP:

```
GET /somedir/page.html HTTP/1.1
Host: www.someschool.edu
Connection: close
User-agent: Mozilla/4.0
Accept-language: fr
```

يمكننا تعلم الكثير بإلقاء نظرة فاحصة على رسالة الطلب البسيطة تلك. أولاً: نلاحظ أن الرسالة مكتوبة بصيغة ASCII كي يتسنى للشخص الذي لديه دراية بالحاسب قراءتها. ثانياً: نلاحظ أن الرسالة تتكون من خمسة سطور، ينتهي كلٌّ منها برمز بداية سطر (carriage return) وتغذية سطر جديد (line feed). أما السطر الأخير من الرسالة فينتهي برمز آخر لبداية سطر وتغذية سطر جديد. بالرغم من أن رسالة الطلب المعينة هذه تتكون من خمسة أسطر، فإن رسالة الطلب عموماً يمكن أن تتضمن سطوراً أكثر أو أقل بحد أدنى سطر واحد. يسمى السطر الأول في رسالة طلب HTTP سطر الطلب (request line)، بينما تسمى السطور اللاحقة سطور الترويسة (header lines). يتكون سطر الطلب من ثلاثة حقول: حقل الأمر (method)، وحقل العنوان (URL)، وحقل رقم إصدار HTTP. يمكن أن يأخذ حقل الأمر عدة قيم مختلفة مثل: GET، POST، HEAD، PUT، DELETE. تستعمل الغالبية العظمى من رسائل طلب HTTP أمر GET والذي يستخدم عندما يريد المتصفّح طلب كائن حيث يضع مسار الكائن المطلوب في حقل العنوان URL. في هذا المثال يطلب المتصفّح الكائن /somedir/page.html (وهو عبارة عن صفحة ويب). أما رقم إصدار HTTP فواضح في هذا المثال أنه عبارة عن HTTP/1.1.

الآن دعنا ننظر إلى سطور الترويسة في المثال السابق. سطر الترويسة

Host: www.someschool.edu

يحدد هذا السطر المضيف الذي يوجد عليه الكائن. قد يبدو أن هذا السطر غير ضروري، حيث إن هناك توصيلة TCP سابقة مع المضيف، لكن كما سنرى في الجزء 2-2-5 فإن المعلومات في سطر الترويسة عن المضيف مطلوبة للذاكرة المخبأة في وكيل الويب – والذي يُعرَف أيضاً بالخادم المفوّض أو بروكسي الويب (web proxy). وبتضّمين السطر

Connection: close

فإن المتصفّح يُخبر الخادم بأنه لا يريد التوصيلات الدائمة (persistent connections)، وإنما يريده أن يغلق التوصيلة بعد إرسال الكائن المطلوب. أما السطر

User-agent: Mozilla/4.0

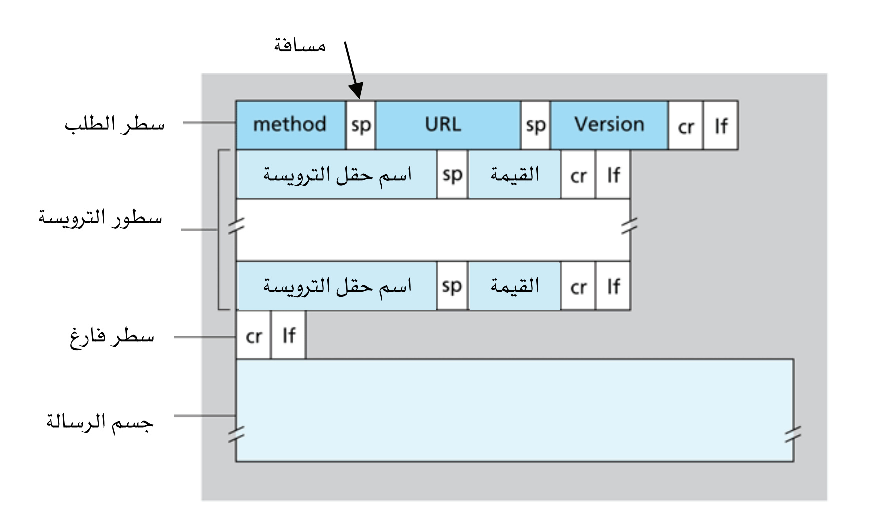
فيحدد نوع المتصفّح الذي أرسل الطلب إلى الخادم، ففي المثال السابق يأخذ القيمة Mozilla/4.0 (أحد المتصفّحات من شركة Netscape). وهذا السطر مفيد لأن الخادم يمكن أن يرسل نسخاً مختلفة من نفس الكائن إلى الأنواع المختلفة من المتصفّحات (كلٌّ منها معنون بنفس العنوان URL). وأخيراً يشير السطر

Accept-language: fr

إلى أن المُستخدِم يفضل استلام نسخة باللغة الفرنسية من الكائن (إذا وجدت على الخادم وإلا فسيرسل الخادم النسخة المعتادة لديه). وهذا السطر من سطور الترويسة هو مجرد مثال للعديد من المفاوضات (negotiations) المتوفرة في HTTP بخصوص المحتوى.

بعد أن استعرضنا المثال السابق دعنا نفحص الآن الصيغة العامة لرسالة طلب كما هي موضحة بالشكل 2-8، وسنرى أن الصيغة العامة قريبة جداً من المثال السابق. ومع ذلك فلعلك لاحظت أنه بعد سطور الترويسة (والسطر الفارغ) يوجد "محتوى الكيان" (entity body)، وهو حقل فارغ عند استخدام GET، ولكنه يُستخدَم مع POST. يستخدم زبون HTTP في أغلب الأحيان POST عندما يملأ المُستخدِم استمارة بيانات، مثلاً عندما يدخل المُستخدِم كلمات الدليلية

(keywords) إلى محرك البحث. في حالة رسالة POST يطلب المُستخدِم أيضاً صفحة ويب من الخادم لكن المحتويات المعينة لصفحة الويب تعتمد على ما أدخله المُستخدِم في حقول البيانات في الاستمارة والتي ترسل ضمن جسم الرسالة.



الشكل 2-8 الصيغة العامة لرسالة طلب HTTP.

وسنكون مقصرين إذا لم نذكر أن رسالة الطلب التي تتولد مع الاستمارة لا يتعين بالضرورة أن تستخدم POST، وإنما في أغلب الأحيان تستخدم طريقة GET وترسل البيانات المدخلة في حقول الاستمارة كجزء من عناوين URL المطلوبة. على سبيل المثال إذا استعملت استمارة بيانات طريقة GET، وبافتراض أن الاستمارة لها حقلان وكانت المُدخلات لهما monkeys وbananas، فإن حقل العنوان URL في الرسالة يصبح

www.somesite.com/animalsearch?monkeys&bananas

ولعلك لاحظت خلال تصفحك اليومي للويب عناوين URL طويلة فمن المحتمل أن تكون من هذا النوع. يشبه الأمر HEAD الأمر GET، فعندما يتلقى خادم طلباً يحتوي على HEAD يرد برسالة HTTP ولكن بدون الكائن المطلوب. وغالباً ما

يستخدم مطوّرو التطبيقات طريقة HEAD عند تعقب وتصحيح الأخطاء أثناء تطوير التطبيق. يُستخدم الأمر PUT في أغلب الأحيان مرتبطاً مع أدوات النشر على الويب، فهو يسمح للمُستخدِم بتحميل كائن إلى مسار معين (مجلد) على خادم ويب معين، كما يُستخدَم بواسطة التطبيقات التي تحتاج لتحميل كائنات على خادمات الويب. أما الأمر DELETE فيسمح لمُستخدِم أو تطبيق بحذف كائن موجود على خادم الويب.

رسائل رد HTTP

نورد فيما يلي رسالة رد HTTP نمطية يمكن أن تكون رسالة رد على رسالة الطلب في المثال الذى تناولناه سابقاً:

```
HTTP/1.1 200 OK
Connection: close
Date: Thu, 07 Jul 2007 12:00:15 GMT
Server: Apache/1.3.0 (Unix)
Last-Modified: Sun, 6 May 2007 09:23:24 GMT
Content-Length: 6821
Content-Type: text/html

(data data data data data ...)
```

دعنا ننظر بعناية في رسالة الرد تلك. تتكون الرسالة من ثلاثة أجزاء: سطر حالة (status line)، وستة سطور ترويسة (header lines)، ثم بعد ذلك جسم الكيان (entity body). إن جسم الكيان هو صُلب الرسالة ويحتوي على الكائن المطلوب نفسه (ممثلاً بـ ... data data data data data). يتكون سطر الحالة من ثلاثة حقول: حقل رقم نسخة بروتوكول HTTP، ورمز الحالة (status code)، ورسالة الحالة المناظرة. في هذا المثال يُشير سطر الحالة إلى أن الخادم يستخدم HTTP/1.1 وأن كل شيء على ما يرام (أي أن الخادم وجد الكيان المطلوب ويقوم بإرساله). دعنا الآن نفحص سطور الترويسة. يستخدم الخادم السطر:

Connection: close

ليخبر الزبون بأنه سيغلق توصيلة TCP بعد إرسال الرسالة.

Date: Thu, 07 Jul 2007 12:00:15 GMT

يُشير إلى وقت وتاريخ إنشاء الرسالة وإرسالها من الخادم. لاحظ أن هذا ليس وقت وتاريخ إنشاء الكائن أو آخر تعديل له، وإنما وقت جلب الخادم للكائن من نظام الملفات لديه، وإدراجه ضمن رسالة الرد، وإرسالها.

Server: Apache/1.3.0 (Unix)

يُشير إلى أن الرسالةَ أنشئت من خادم ويب Apache، وهي مماثلة لسطر العنوان

User-agent:

والتي تحدد نوع المتصفّح في رسالة طلب HTTP. أما السطر:

Last-Modified: Sun, 6 May 2007 09:23:24 GMT

فيُشير إلى وقت وتاريخ إنشاء الكائن أو آخر تعديل له، وسوف نتناوله قريباً بتفصيل أكثر نظراً لأهميته للتخزين المؤقت للكائن في الذاكرة المخبأة لدى كل من الزبون المحلي وخادمات الذاكرة المخبأة على الشبكة. أما السطر

Content-Length: 6821

فيُشير إلى حجم الكائن المُرسَل (بالبايتات). والسطر:

Content-Type: text/html

يُشير إلى أن الكائن المتضمن في الرسالة (جسم الكيان) نص HTML، (لاحظ أن نوع الكائن يحدد رسمياً بهذا السطر وليس بامتداد الملف).

بعد أن استعرضنا مثالاً دعنا الآن نفحص الصيغة العامة لرسالة الرد كما هي موضحة في الشكل 2-9، والتي تتطابق مع المثال السابق. دعنا نضيف بضع كلمات إضافية عن رموز الحالة (status code) وعباراتها (phrases) المناظرة. يُشير رمز الحالة والعبارة المصاحبة له إلى نتيجة الطلب. فيما يلي بعض رموز الحالة والعبارات المصاحبة لها:

- 200 OK

تعني نجاح العملية وإرسال المعلومات المطلوبة في الرسالة.

- 301 Moved Permanently

تعني أن الكائن المطلوب تم نقله بصفة دائمة للموقع المحدد في السطر Location من سطور الترويسة في رسالة الرد. سيقوم برنامج الزبون باسترجاع الكائن من الموقع الجديد تلقائياً.

- 400 Bad Request

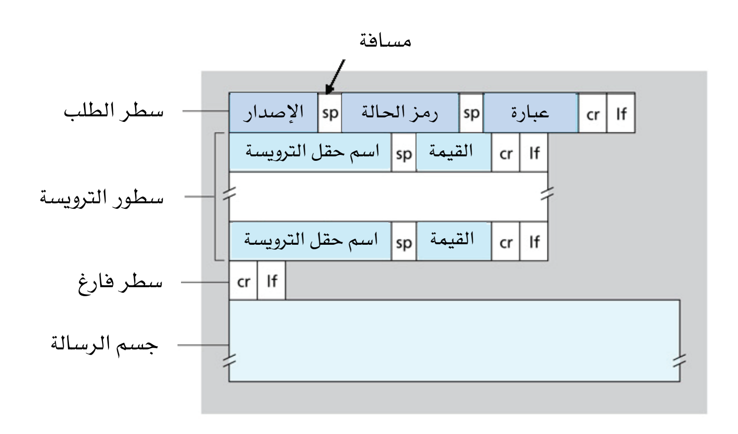
رسالة خطأ عامة تعني أن الخادم لم يفهم الطلب.

- 404 Not Found

تعني أن الكائن المطلوب غير موجود على الخادم.

- 505 HTTP Version Not Supported

تعني أن نسخة HTTP المُستخدَمة غير مدعومة من قِبَل الخادم.



الشكل 2-9 الصيغة العامة لرسالة رد HTTP.

هل تريد أن ترى رسالة رد HTTP حقيقية؟ هذا الأمر يوصى به بشدة كما أنه سهل المنال! أولاً استخدم telnet للوصول إلى خادم الويب المفضل لديك، ثم

اكتب رسالة طلب تتكون من سطر واحد لكائن ما على الخادم، على سبيل المثال عن طريق واجهة الأوامر (command prompt) اكتب:

```
telnet cis.poly.edu 80
GET /~ross/ HTTP/1.1
Host: cis.poly.edu
```

ثم اضغط على زر الإدخال مرتين بعد السطر الأخير. يتم فتح توصيلة TCP إلى منفذ 80 على المضيف cis.poly.edu، وبعد ذلك تُرسَل رسالة طلب HTTP. وكنتيجة لذلك سترى رسالة رد تتضمن ملف HTML الأساسي للصفحة الرئيسية للأستاذ Ross بجامعة بولي تكنك. أما إذا أردت أن ترى فقط سطور الترويسة لرسالة HTTP دون الحصول على الكائن نفسه فاستخدم HEAD بدلاً من GET. وأخيراً استبدل /~ross/ بـ /~banana/ لترى رسالة خطأ في الرد.

في هذا الجزء ناقشنا عدداً من سطور الترويسة التي تستخدم ضمن رسائل طلب ورد HTTP. وتُعرِّف مواصفات HTTP الكثير من سطور الترويسة والتي يمكن أَن تقوم بإدخالها المتصفّحات وخادمات الويب وخادمات الذاكرة المخبأة على الشبكة. لقد تم تغطية عدد محدود فقط من مجموع سطور الترويسة، وسوف نغطي بضعة سطور أخرى فيا بعد وعدداً قليلاً آخر عندما نناقش استخدام الويب للذاكرة المخبأة في الجزء 2-2-5. توجد مناقشة شاملة وسهلة القراءة لنظام HTTP تتضمن رموز الحالة وسطور الترويسة في [Krishnamurty 2001]؛ راجع أيضاً [Luotonen 1998] لاستعراض الموضوع من وجهة نظر مطوّري التطبيقات.

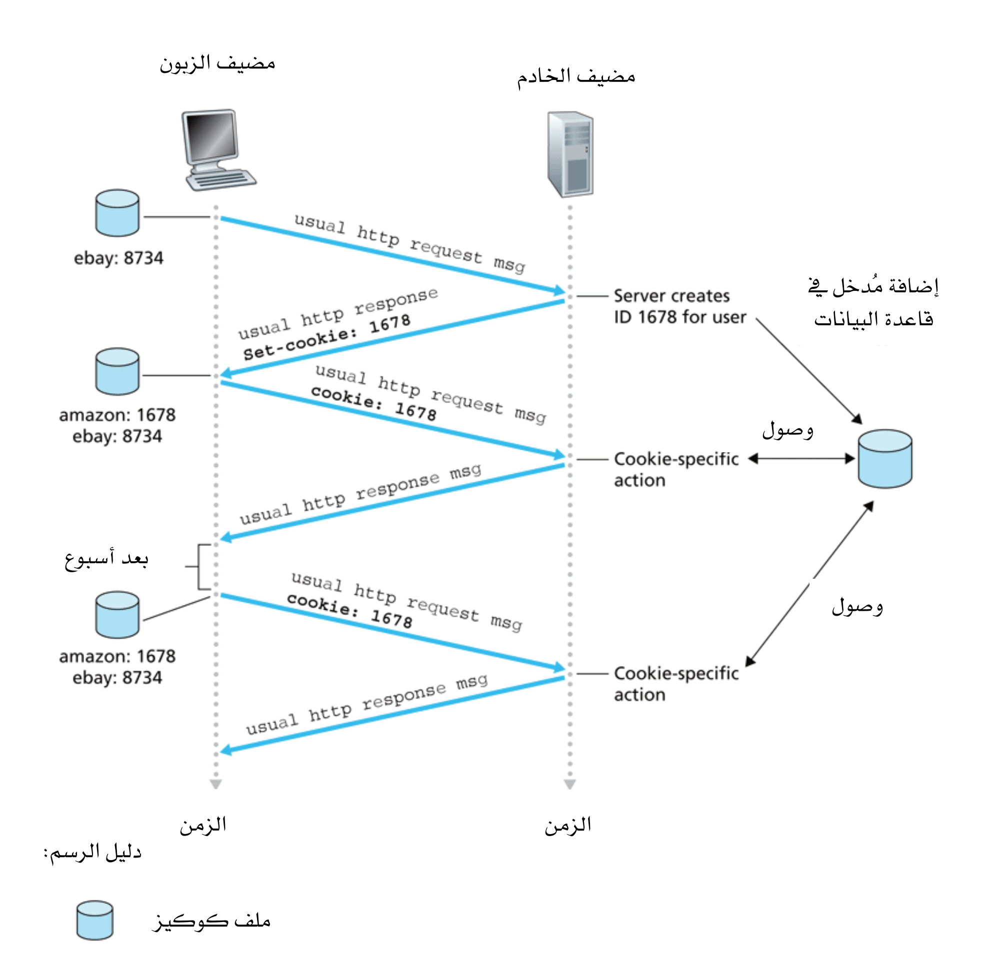
كيف يقرر متصفّحٌ ما سطور الترويسة التي يُضمِّنها في رسالة الطلب؟ وكيف يقرر خادم ما على الويب سطور الترويسة التي يُضمِّنها في رسالة الرد؟ يتم ذلك طبقاً لنوع المتصفّح ورقم إصداره (على سبيل المثال لا يولد متصفّح HTTP/1.0 أياً من سطور الترويسة الخاصة بمتصفّح HTTP/1.1)، وكذلك طبقاً لتهيئة أو إعداد المُستخدِم للمتصفّح (على سبيل المثال اللغة المُفضَّلة)، وأيضاً إذا ما كان

لديه حالياً نسخة من الكائن في ذاكرته المخبأة ولكنها قد تكون منتهية التاريخ. كذلك تتصرف خادمات الويب بنفس الطريقة: هناك العديد من المنتجات والإصدارات والإعدادات المختلفة، والتي تؤثر جميعها على سطور الترويسة المُتضمَّنة في رسائل الرد.

2-2-4 التفاعل بين المُستخدِم والخادم : الكوكيز (Cookies)

سبق أن ذكرنا أن خادم HTTP بلا ذاكرة للحالة (stateless)، مما يبسط تصميمه ويسمح للمهندسين بتطوير خادمات ويب عالية الأداء يمكن أن تعالج آلاف توصيلات TCP في نفس الوقت. ومع ذلك فغالباً ما يرغب موقع الويب في أن تكون لديه القدرة على تمييز المُستخدِمين، إما لأن الخادم يرغب في قصر الوصول إليه على مُستخدِمين بعينهم، أو لأنه يرغب في ربط خدمة المحتوى بهوية المُستخدِم. ولهذه الأغراض يستخدم بروتوكول HTTP الكوكيز (وهي مُعرّفة في طلب التعليقات RFC 2965) والتي تسمح لمواقع الويب بمتابعة المُستخدِمين. تستخدم أكثر مواقع الويب التجارية الرئيسة الكوكيزهذه الأيام.

كما هو موضح في الشكل 2-10 تتضمن تقنية الكوكيز أربعة مكوّنات: (1) سطر ترويسة للكوكيز في رسالة رد HTTP، (2) سطر ترويسة للكوكيز في رسالة طلب HTTP، (3) ملف كوكيز مُخزّن على النظام الطرفي للمُستخدِم ومُدار بمتصفّح المُستخدِم، (4) قاعدة بيانات في الخلفية على موقع الويب. بالاستعانة بشكل 2-10 دعنا نتتبع مثالاً يوضح كيفية عمل الكوكيز. افرض أن سوزان التي تستخدم دائماً المتصفّح إنترنت إكسبلورر من جهاز حاسبها الشخصى في بيتها، زارت موقع Amazon.com للمرة الأولى، ودعنا نفرض أنه قد سبق لها أن زارت موقع eBay. عندما يجيء الطلب إلى خادم ويب Amazon يقوم الخادم بإنشاء رقم فريد لتعريف هوية المُستخدِم ويضيف مدخلاً جديداً إلى قاعدة بياناته المفهرسة بذلك الرقم التعريفي. بعد ذلك يرد خادم ويب Amazon على متصفّح سوزان مُضمِّناً في الرد سطر الترويسة Set-cookie والذي يحتوي على الرقم التعريفي، على سبيل المثال: Set-cookie: 1678.



الشكل 2-10 الاحتفاظ بحالة المُستخدِم عن طريق الكوكيز.

وعندما يتسلم متصفّح سوزان رسالة الرد ، يرى ذلك السطر.عندئذ يضيف المتصفّح سطراً إلى ملف الكوكيز الخاص الذي يحتفظ به لديه يتضمن اسم مضيف الخادم وذلك الرقم التعريفي. لاحظ أن ملف الكوكيز يحتوي أيضاً على مُدخل لموقع eBay منذ أن زارت سوزان ذلك الموقع في الماضي. وبينما تواصل سوزان تَصفُّح موقع Amazon ، ففي كل مرة تطلب صفحة ويب، يستشير متصفّحها ملف

الكوكيز ويستخلص الرقم التعريفي لذلك الموقع، ويضع سطر ترويسة لرسالة الطلب يتضمن الرقم التعريفي. وبالتحديد ستتضمن كل طلباتها المُرسَلة إلى خادم Amazon سطر الترويسة:

Cookie: 1678

وبهذا الشكل يستطيع خادم Amazon تعقب نشاط سوزان في موقع Amazon بالرغم من أن موقع ويب Amazon لا يعرف بالضرورة اسم سوزان، وإنما يعرف بالضبط أي الصفحات زارها المُستخدِم رقم 1678، وبأي ترتيب وفي أي وقت! يستخدم موقع Amazon الكوكيز لتقديم خدمة عربة التسوّق (shopping cart) – حيث يمكن له الاحتفاظ بقائمة بكل البنود التي تنوي سوزان شراءها، لكي تتمكن من دفع ثمنهم مرة واحدة في نهاية الجلسة. وإذا عادت سوزان إلى موقع Amazon لاحقاً (بعد أسبوع مثلاً)، سيواصل متصفّحها وضع سطر الكوكيز

Set-cookie: 1678

في رسائل الطلب. كما يمكن أيضاً أن يقترح موقع Amazon بعض المنتجات لسوزان حسب صفحات الويب التي زارتها في الماضي. أمّا إذا سجّلت سوزان نفسها أيضاً بموقع Amazon بإعطاء اسمها بالكامل، وعنوان بريدها الإلكتروني، وعنوانها البريدي، ومعلومات بطاقتها الائتمانية – فإن موقع Amazon يمكن عندئذ أن تضيف تلك المعلومات إلى قاعدة بياناتها وبذلك يرتبط اسم سوزان بالرقم التعريفي لها (وكل الصفحات التي زارتها على الموقع في الماضي!). وهذه هي كيفية توفير موقع Amazon ومواقع التجارة الإلكترونية الأخرى خدمة "التسوّق بنقرة واحدة". فعندما تختار سوزان شراء مادة أثناء زيارة لاحقة، لن تحتاج إلى إعادة إدخال اسمها أو رقم بطاقة الائتمان لها أو عنوانها.

من هذه المناقشة نرى أن الكوكيز cookies يمكن أن تُستعمَل لتمييز المُستخدِم. فأول مرة يزور المُستخدِم موقعاً ما يُعطى رمزاً تعريفياً (مثلاً اسمه أو اسمها). وأثناء الجلسات اللاحقة يُرسِل المتصفّح سطر الكوكيز إلى الخادم، وبذلك يستطيع الخادم تمييز المُستخدِم. وهكذا يمكن استعمال الكوكيز

لإنشاء طبقة الجلسة للمُستخدِم فوق بروتوكول HTTP (الذى لا يتذكر الحالة). فعلى سبيل المثال عندما يدخل المُستخدِم إلى تطبيق البريد الإلكتروني على الإنترنت (كـ hotmail مثلاً) يُرسِل المتصفّح معلومات الكوكيز إلى الخادم مما يسمح للخادم بتمييز المُستخدِم طوال جلسة اتصاله بالتطبيق.

على الرغم من أن الكوكيز غالباً ما يبسّط عملية التسوّق من خلال الإنترنت فهناك جدل حولها، حيث يمكن أيضاً أن تضر بالخصوصية (privacy). فكما لاحظنا سابقاً يمكن لموقع الويب – باستخدام الكوكيز مع معلومات حساب المُستخدِم – أن يعرف الكثير عن المُستخدِم ثم يبيع تلك المعلومات إلى طرف ثالث. يتضمن موقع [Cookie Central 2007] معلومات شاملة عن الجدل القائم حالياً حول الكوكيز.

5-2-2 ذاكرة الويب المخبأة (Web Caching)

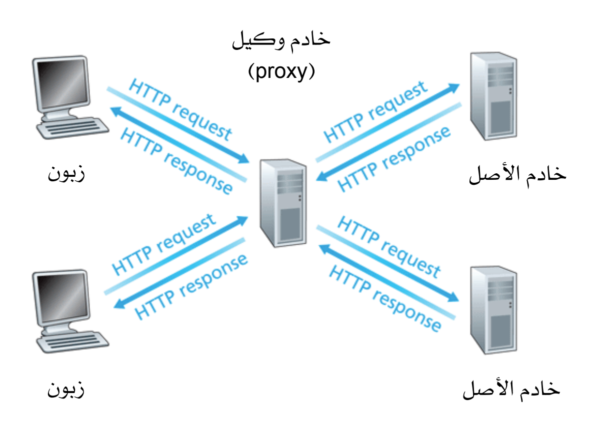
ذاكرة الويب المخبأة والتي تعرف أيضاً بالخادم المفوّض أو الوكيل (proxy server) هي أحد كيانات الشبكة التي تخدم طلبات HTTP نيابةً عن خادم الويب الأصلي، ولها أقراص تخزين (disk storage) خاصة بها تحتفظ فيها بنسخ من الكائنات المطلوبة حديثاً. كما يوضح الشكل 2-11، يمكن ضبط المتصفّح بحيث يُوجّه كل طلبات HTTP من المُستخدِم أولاً إلى ذاكرة الويب المخبأة. بمجرد إتمام عملية الضبط تلك، سيوجّه المتصفّح كل طلب أولاً إلى ذاكرة الويب المخبأة. كمثال افرض أن المتصفّح يطلب الكائن:

http:://www.someschool.edu/campus.gif

فسيحدث ما يلي:

1. يُنشئ المتصفّح توصيلة TCP إلى ذاكرة الويب المخبأة ويُرسِل إليها طلب HTTP للكائن.
2. تفحص ذاكرة الويب المخبأة لترى ما إذا كان لديها نسخة من الكائن مخزنة محلياً، فإذا وجدتها فإنها ترسلها ضمن رسالة رد HTTP إلى المتصفّح.

3. أما إذا لم يكن لديها الكائن، تفتح ذاكرة الويب المخبأة توصيلة TCP إلى الخادم الأصلي (المصدر) أي www.someschool.edu، وبعد ذلك ترسل ذاكرة الويب المخبأة طلب HTTP للكائن عبر توصيلة TCP إلى ذلك الخادم . وبعد استلام الخادم الأصلي لهذا الطلب، يُرسِل الكائن ضمن رسالة رد HTTP إلى ذاكرة الويب المخبأة.
4. عندما تتسلم ذاكرة الويب المخبأة الكائن تقوم بتخزين نسخة منه في وحدة التخزين المحلي لديها وتُرسِل نسخة ضمن رسالة رد HTTP إلى متصفّح الزبون (على توصيلة TCP الحالية بين متصفح الزبون وذاكرة الويب المخبأة).



الشكل 2-11 الزبائن تطلب كائنات عن طريق ذاكرة الويب المخبأة.

لاحظ أن الذاكرة المخبأة تعمل كخادم وكزبون في نفس الوقت، فعندما تستلم طلبات من المتصفّح وتُرسل الردود إليه، فهي عندئذ تعمل كخادم، وعندما ترسل بالطلبات إلى الخادم الأصلي وتستلم الردود منه فإنها تعمل عند ذاك كزبون. عادةً ما تُشتَرى ذاكرة الويب المخبأة وتُركَّب من قِبَل موفّر خدمة الإنترنت. على سبيل المثال قد تُركِّب جامعةٌ ما ذاكرة مخبأة على شبكة الحرم الجامعي، وتضبط كل متصفّحات الحرم الجامعي لتشير إلى تلك الذاكرة المخبأة. كما يمكن أن يُركِّب موفّر رئيس لخدمة الإنترنت السكني

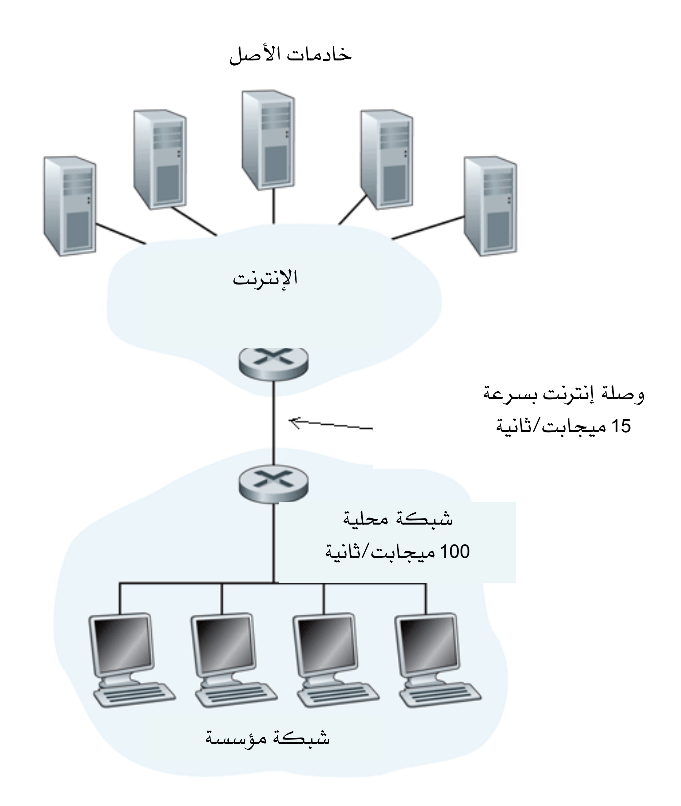
(residential ISP) (مثل AOL) واحدةً أو أكثر من تلك الذاكرات على شبكته، ويضبط متصفّحاته لتشير إليها.

يرجع انتشار استخدام ذاكرة الويب المخبأة في الإنترنت لسببين. أولاً: يمكن أن تقلل ذاكرة الويب المخبأة وقت الرد على طلب الزبون بشكلٍ ملحوظ، خاصة إذا كان الحيّز الترددي الذي يمثل عنق الزجاجة بين الزبون والخادم الأصلي أقل بكثير من الحيّز الترددي الذي يمثل عنق الزجاجة بين الزبون والذاكرة المخبأة. وإذا كانت الوصلة سريعة بين الزبون والذاكرة المخبأة (كما هو الحال في الغالب) وكانت الذاكرة المخبأة تحوي الكائن المطلوب، فحينئذ سيكون بوسع الذاكرة المخبأة تسليم الكائن بسرعة إلى الزبون. ثانياً: كما سنوضح قريباً بمثال يمكن أن تؤدي الذاكرة المخبأة إلى خفض حركة البيانات على وصلة اتصال مؤسسة (شركة أو جامعة مثلاً) بالإنترنت بشكلٍ جوهري. وبخفض حركة البيانات فإن المؤسسة (الشركة أو الجامعة) لن تكون مضطرة لترقية (توسّعة) (upgrade) الحيّز الترددي لوصلتها بالإنترنت بسرعة، مما يؤدي إلى خفض التكلفة. وعلاوةً على ذلك يمكن أن تقلل ذاكرة الويب المخبأة حركة بيانات الويب في الإنترنت ككل بشكلٍ ملحوظ مما يُحسِّن أداء كل التطبيقات.

ولفهم فوائد الذاكرة المخبأة بعمق دعنا نأخذ مثالاً ضمن سياق الشكل 2-12 والذي يبين شبكتين (شبكة المؤسّسة وبقيّة شبكة الإنترنت العامة). شبكة المؤسّسة هي شبكة اتصالات محلية بسرعة عالية (High-Speed LAN). يتصل الموجّه على تلك الشبكة بموجّه الإنترنت بوصلة سرعتها 15 ميجابت/ثانية. ترتبط خادمات المصدر بالإنترنت وهي موزّعة في جميع أنحاء الكرة الأرضية. افترض أن الحجم المتوسط للكائن هو 1 ميجابت، وأن المعدل المتوسط لطلبات الحصول على الكائنات من متصفّحات المؤسّسة إلى خادمات الأصل هو 15 طلب في الثانية. وافترض أن رسائل طلب HTTP صغيرة جداً (بدرجة يمكن إهمالها)، وبالتالي لن تُولّد أي حركة مرور تُذكر للبيانات في الشبكات أو في الوصلة من موجّه المؤسّسة إلى موجّه الإنترنت. افترض أيضاً أن الوقت الذي يمر من اللحظة التي يقوم فيها الموجه على جانب الإنترنت من الوصلة في الشكل 2-12 بتوجيه طلب

HTTP (ضمن وحدة بيانات IP) إلى أن يتلقى الرد (عادة على شكل عدة وحدات بيانات IP) يعادل ثانيتين في المتوسط. وبطريقة غير رسمية يطلق على زمن التأخير هذا "تأخير الإنترنت".

زمن الاستجابة الكلي (أي الفترة الزمنية من حين طلب المتصفّح الكائن حتى حصوله عليه) هو مجموع زمن التأخير على الشبكة المحلية وزمن التأخير على وصلة التوصل بالإنترنت (أي الوصلة بين الموجّهين) وزمن التأخير على الإنترنت. دعنا نجري بعض الحسابات التقريبية لتقدير ذلك الزمن.



الشكل 2-12 عنق الزجاجة بين شبكة مؤسّسة والإنترنت.

تبلغ كثافة المرور على الشبكة المحلية (راجع الجزء 1-4-2):

(15 requests/sec) × (1 Mbits/request) / (100 Mbps) = 0.15

بينما تبلغ كثافة المرورعلى الوصلة من موجّه الإنترنت إلى موجّه المؤسّسة:

(15 requests/sec) × (1 Mbits/request) / (15 Mbps) = 1

تؤدي كثافة المرور 0.15 على شبكة الاتصالات المحلية إلى تأخير يعادل (على الأغلب) عشرات الميللي ثانية، وبالتالي يمكننا إهمال التأخير على شبكة الاتصالات المحلية. ولكن (كما نوقش في الجزء 1-4-2) مع اقتراب كثافة المرور من واحد (كما هو الحال على الوصلة بين الموجّهين في الشكل 2-12) فإن التأخيرعلى الوصلة يصبح كبيراً جداً وينمو بدون حد. وهكذا فإن الزمن المتوسط لتحقيق الطلبات سيصبح في حدود الدقائق (إن لم يكن أكثر)، وهذا غير مقبول لمُستخدِمي المؤسّسة، مما يُحتّم إيجاد حل بديل.

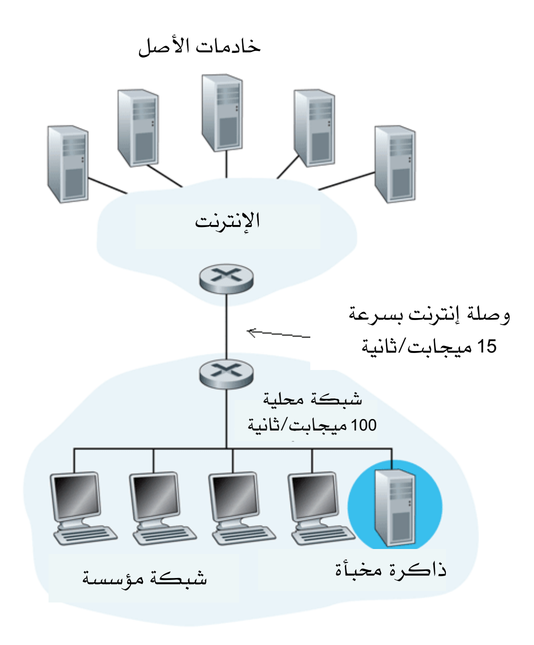
قد يكون أحد الحلول الممكنة زيادة سرعة وصلة التوصل بالإنترنت إلى 100 ميجابت/ثانية مثلاً بدلاً من 15 ميجابت/ثانية. سيؤدي ذلك إلى تخفيض كثافةَ المرور على وصلة التوصل إلى 0.15 والتي تؤدي إلى تأخير بسيط بين الموجّهين. في هذه الحالة سيكون وقت الرد الكلي ثانيتين تقريباً، أي مساوياً لتأخير الإنترنت. لكن هذا الحل يعني أيضاً أن المؤسّسة يجب أن تُرقّي الوصلة التي تربطها بالإنترنت من 15 ميجابت/ثانية إلى 100 ميجابت/ثانية وهو حل مكلّف.

لننظر بعين الاعتبار لحل بديل لا يتطلب ترقيّة لوصلة الإنترنت، ولكن (بدلاً من ذلك) يتضمن تركيب ذاكرة ويب مخبأة على شبكة المؤسّسة، كما هو موضّح في الشكل 2-13. يترواح معدل إصابة الهدف (hit rate) (نسبة الطلبات التي تتمكن الذاكرة المخبأة من تلبيتها) ما بين 0.2 إلى 0.7 عملياً. ومن أجل الإيضاح دعنا نفترض أن الذاكرة المخبأة توفر معدل إصابة للهدف يعادل 0.4 لتلك المؤسّسة. ولأن الزبائن والذاكرة المخبأة متصلون بنفس شبكة الاتصالات المحلية السريعة فإن 40٪ من الطلبات تقريباً ستُلبى فوراً (مثلاً خلال 10 ميللي ثانية) من

الذاكرة المخبأة. أما نسبة الـ 60٪ المتبقية من الطلبات فلا يزال من الضروري إرسالها إلى خادمات الأصل. لكن نظراً لأن 60٪ فقط من الكائنات المطلوبة تعبر الآن وصلة الإنترنت، فإن كثافة المرورعلى الوصلة تنخفض من 1.0 إلى 0.6. وطبيعي أن كثافة مرور أقل من 0.8 تناظر تأخيراً صغيراً (مثلاً بضع عشرات ميللي ثانية) على الوصلة بسرعة 15 ميجابت/ثانية. هذا التأخير بسيط مقارنة بالثانيتين (زمن تأخير الإنترنت). ولذا فإن متوسط زمن التأخير تحت هذه الظروف يساوي:

$$0.4 \times (0.01 \text{ seconds}) + 0.6 \times (2.01 \text{ seconds})$$

وهذا يتجاوز 1.2 ثانية بقليل. وهكذا يوفر هذا الحل الثاني زمن استجابة أقل من الحل الأول، كما أنه لا يتطلب من المؤسّسة ترقية وصلة الإنترنت لديها. وبالطبع يجب على المؤسّسة شراء وتركيب ذاكرة ويب مخبأة، غير أن تكلفة ذلك منخفضة - فالعديد من ذاكرات الويب المخبأة تستخدم برامج عامة تعمل على حاسبات شخصية رخيصة.



الشكل 2-13 إضافة ذاكرة مخبأة إلى شبكة المؤسّسة.

6-2-2 أمر GET الشّرطي

رغم أن استخدام ذاكرة مخبأة يمكن أن يُخفّض أوقات الاستجابة المحسوسة للمُستخدِم، إلا أنها تأتي بمشكلة جديدة (فَقَد تتقادم نسخة الكائن الموجود على الذاكرة المخبأة). وبمعنى آخر ربما يكون قد جرى تعديل على الكائن الموجود على خادم الويب الأصلي بعد وضع نسخة من ذلك الكائن في الذاكرة المخبأة لدى الزبون. لحسن الحظ يتضمن بروتوكول HTTP آلية تسمح للذاكرة المخبأة بالتحقق من أن الكائنات حديثة حتى اللحظة. هذه الآلية هي أمر GET الشّرطي. تسمّى رسالة طلب HTTP برسالة GET الشّرطية إذا (1) استعملت رسالة الطلب أمر GET، (2) تضمنت الرسالة سطر الترويسة:

If-Modified-Since:

ولتوضيح كيفية عمل أمر GET الشّرطي دعنا نستعرض هذا المثال. أولاً: يُرسل خادم الذاكرة المخبأة نيابةً عن المتصفّح رسالة طلب إلى خادم الويب:

```
GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
```

ثانياً: يُرسِل خادم الويب رسالة رد بالكائن المطلوب إلى الذاكرة المخبأة:

```
HTTP/1.1 200 OK
Date: Thu, 7 Jul 2007 15:39:29
Server: Apache/1.3.0 (Unix)
Last-Modified: Mon, 4 Jul 2007 09:23:24
Content-Type: image/gif

(data data data data data ...)
```

تُرسِل الذاكرة المخبأة الكائن إلى المتصفّح الطالب لكن أيضاً مع الاحتفاظ بنسخة من الكائن محلياً. كما تخزن الذاكرة المخبأة تاريخ آخر تعديل سويّة مع الكائن. ثالثاً: بعد أسبوع من ذلك التاريخ يطلب متصفّح آخر نفس

الكائن عن طريق الذاكرة المخبأة، والكائن ما زال في الذاكرة المخبأة. نظراً لاحتمال كون هذا الكائن قد عُدِّل في خادم الويب خلال ذلك الأسبوع، تقوم الذاكرة المخبأة بالتأكد من ذلك عن طريق إصدار رسالة GET الشّرطية. وبالتحديد تُرسِل الذاكرة المخبأة:

GET /fruit/kiwi.gif HTTP/1.1
Host: www.exotiquecuisine.com
If-Modified-Since: Wed, 4 Jul 2007 09:23:24

لاحظ أن قيمة السطر If-Modified-Since تساوي بالضبط قيمة Last-Modified التي أرسلها الخادم قبل أسبوع مضى. يطلب هذا الامر الشّرطي من الخادم أن يُرسِل الكائن من جديد إذا كان قد عُدِّل منذ التاريخ المحدد. افترض أن الكائن لم يُعدَّل منذ 4 يوليو/تموز 2007 09:23:24. حينئذ يُرسِل خادم الويب رسالة رد إلى الذاكرة المخبأة:

HTTP/1.1 304 Not Modified
Date: Thu, 14 Jul 2007 15:39:29
Server: Apache/1.3.0 (Unix)

(empty entity body)

وكما نرى في الرد على رسالة GET الشّرطية، ما زال خادم الويب يُرسِل رسالة رد لا تتضمن الكائن المطلوب. فتضمين الكائن المطلوب سيؤدي إلى إهدار الحيّز الترددي المتاح، وزيادة زمن الاستجابة المحسوس للمُستخدِم (خصوصاً إذا كان حجم الكائن المطلوب كبيراً). لاحظ أن رسالة الرد الأخيرة هذه تحتوي على

344 Not Modified

في سطر الحالة (أي لم يحدث تعديل للكائن المطلوب منذ التاريخ المذكور)، وهذا يخبرالذاكرة المخبأة بأن بإمكانها المُضي قُدُماً في إرسال نسخة الكائن الموجودة عليها إلى المتصفّح الطالب.

بهذا تنتهي مناقشتنا لبروتوكول HTTP (أول بروتوكولات الإنترنت التي ندرسها بالتفصيل، وهو أحد بروتوكولات طبقة التطبيقات). ولقد استعرضنا من

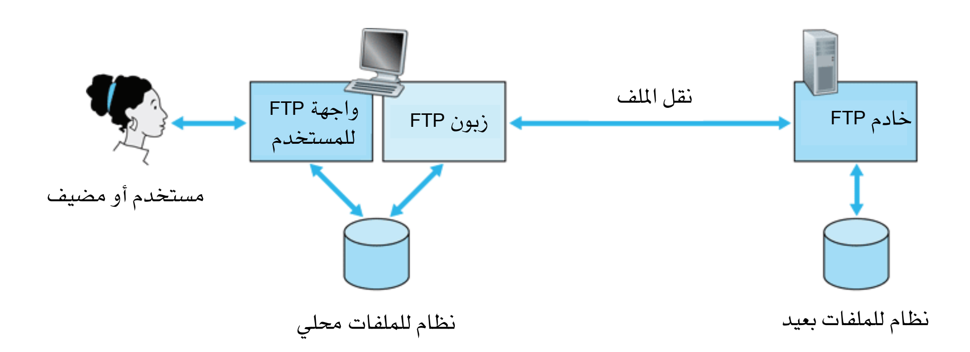
خلال تلك المناقشة صيغ رسائل HTTP والأعمال التي يقوم بها كلٌّ من زبون وخادم الويب عند إرسال الرسائل واستلامها. وتعرّضنا بعض الشيء للبنية التحتية لتطبيق الويب، بما في ذلك الذاكرة المخبأة والكوكيز وقواعد البيانات الخلفية والتي ترتبط جميعها بشكلٍ أو بآخر ببروتوكول HTTP.

2-3 نقل الملفات باستخدام بروتوكول FTP

في جلسة FTP المعتادة يجلس المُستخدِم أمام حاسبه (المضيف المحلي) ويريد نقل ملفات من مضيف بعيد وإليه. ولكي يتمكن المُستخدِم من الوصول للحاسب البعيد يجب عليه أولاً أن يُدخل تعريف المُستخدِم وكلمة المرور (كلمة السّر) الخاصين به. وبعد إعطاء معلومات التفويض هذه يمكنه نقل ملفات من نظام الملفات المحلي إلى نظام الملفات البعيد والعكس بالعكس. وكما هو موضح في الشكل 2-14 يتعامل المُستخدِم مع بروتوكول FTP من خلال برنامج زبون المُستخدِم للبروتوكول. يقوم المُستخدِم أولاً بإدخال اسم المضيف للجهاز البعيد وبالتالي تقوم عملية زبون FTP في المضيف المحلي بإنشاء توصيلة TCP مع عملية الخادم في المضيف البعيد. عندئذ يُدخِل المُستخدِم تعريف المُستخدِم وكلمة السّر الخاصين به، فتُرسَل عبر توصيلة TCP كجزء من أوامر FTP إلى عملية الخادم. بمجرد تأكد الخادم من صلاحية التعامل مع المُستخدِم يمكن للمُستخدِم أن ينسخ ملفاً أو أكثر من الملفات المخزّنة في نظام الملفات المحلي إلى نظام الملفات البعيد (أو العكس).

يعتبر HTTP وFTP بروتوكولات لنقل الملفات ولهما العديد من الخصائص المشتركة. على سبيل المثال كلاهما بروتوكول لطبقة التطبيقات يعمل فوق بروتوكول TCP. ومع ذلك توجد بعض الاختلافات المهمة بينهما. إن الاختلاف الأكثر وضوحاً هو أن FTP يستخدم توصيلتي TCP متوازيتين لنقل ملف ما، يُطلق على الأولى توصيلة التحكم (control connection) وعلى الثانية توصيلة البيانات (data connection). تُستخدَم توصيلة التحكم لإرسال معلومات التحكم بين المضيفين – كتعريف المُستخدِم، وكلمة السّر، وأوامر تغيير مجلد الملفات البعيد

(remote directory)، والأوامر الخاصة بـ "تحميل" أو"تنزيل" الملفات. أما توصيلة البيانات فتستخدم لإرسال الملف الحقيقي. وبسبب استخدام FTP توصيلة تحكم مستقلة يقال إن FTP يرسل معلومات التحكم خارج النطاق (out-of-band). سنرى في الفصل السابع أن بروتوكول RTSP (والذي يُستخدَم للتحكم في تشغيل الوسائط المتعددة كتسجيلات الصوت والفيديو) يُرسِل معلومات التحكم أيضاً خارج النطاق.



الشكل 2-14 نقل ملفات بين نظامي الملفات المحلي والبعيد باستخدام FTP.

أما HTTP (كما تتذكّر) فيُرسِل سطور الترويسة (header lines) لرسائل الطلب والرد على نفس توصيلة TCP التي تحمل الملف المنقول، ولهذا السبب يُقال إن HTTP يرسل معلومات التحكم داخل النطاق (in-band). في الجزء التالي سوف نرى أن بروتوكول SMTP (البروتوكول الأساسي للبريد الإلكتروني) يرسل معلومات التحكم أيضاً داخل النطاق. يوضح الشكل 2-15 توصيلة التحكم وتوصيلة البيانات لبروتوكول FTP.



الشكل 2-15 توصيلتا التحكم والبيانات لبروتوكول FTP.

عندما يبدأ المُستخدِم جلسة FTP مع مضيف بعيد فإن جانب زبون FTP (المُستخدِم) يبدأ بإنشاء توصيلة تحكم TCP إلى الخادم (المضيف البعيد) على منفذ الخادم رقم 21. يُرسِل جانب زبون FTP تعريف المُستخدِم وكلمة السرّ على تلك التوصيلة، كما يُرسِل جانب زبون FTP أيضاً على توصيلة التحكم أوامر لتغيير الدليل البعيد. وعندما يستلم جانب الخادم أمراً على توصيلة التحكم لنقل ملف (سواءً إلى المضيف البعيد أو منه)، يبدأ جانب الخادم توصيلة بيانات TCP إلى الزبون. يُرسِل FTP ملفاً واحداً على توصيلة البيانات، وبعدها يغلق توصيلة البيانات تلك. وإذا أراد المُستخدِم أثناء نفس الجلسة نقل ملف آخر، يفتح FTP توصيلة بيانات أخرى. وهكذا فإن FTP يُبقِي توصيلة التحكم مفتوحة طوال مدّة جلسة المُستخدِم، لكنه يُنشئ توصيلة بيانات جديدة لكل ملف يتم نقله أثناء الجلسة (أي أن توصيلات البيانات في FTP غير دائمة (non-persistent).

يجب أن يحتفظ الخادم بمعلومات عن حالة المُستخدِم أثناء الجلسة. وبشكلٍ خاص يجب أن يقرن الخادم توصيلة التحكم مع حساب مُستخدِم (user account) معين، كما يجب أن يتتبع الخادم دليل الملفات الحالي للمُستخدِم أثناء تجوّله خلال شجرة الدليل (الفهرس) على المضيف البعيد. يلاحظ أن متابعة تلك المعلومات الرسمية لكل جلسة مستمرة يُحِد بشكلٍ ملحوظ من العدد الكلي للجلسات التي يمكن أن يقوم بها بروتوكول FTP في نفس الوقت. تذكّر في المقابل أن بروتوكول HTTP عديم الحالة (stateless)، بمعنى أنه لا يتتبع أية حالة للمُستخدِم.

2-3-1 أوامر وردود بروتوكول FTP

دعنا ننهي هذا الجزء بمناقشة قصيرة لبعض أوامر وردود FTP الأكثر شيوعاً. ترسل الأوامر (من الزبون إلى الخادم)، والردود (من الخادم إلى الزبون) عبر توصيلة التحكم في صيغة ASCII بطول 7 بتات. وهكذا فإن أوامر FTP – مثله في ذلك مثل HTTP – يمكن قراءتها من قِبَل الناس. ولفصل الأوامر المتعاقبة يُستخدَم رمز بداية سطر (carriage return) ورمز تغذية سطر جديد (line feed).

يتكون كل أمر من أربعة حروف ASCII كبيرة كما أن بعض الأوامر لها معاملات اختيارية. وفيما يلي وصف لبعض الأوامر الأكثر شيوعاً:

- USER username

يُستخدَم لإرسال تعريف المُستخدِم إلى الخادم.

- PASS password

يُستخدَم لإرسال كلمة السّر الخاصة بالمُستخدِم إلى الخادم .

- LIST

يُستخدَم لطلب إرسال قائمة بكل الملفات الموجودة في الدليل البعيد الحالي من الخادم. تُرسَل قائمة الملفات على توصيلة بيانات (جديدة وغير دائمة) بدلاً من توصيلة TCP الخاصة بالتحكم.

- RETR filename

يُستخدَم لاستجلاب ملف من الدليل الحالي على المضيف البعيد. عند تلقي هذا الأمر يُنشئ المضيف البعيد توصيلة بيانات ويرسل الملف المطلوب عليها.

- STOR filename

يُستخدَم لتخزين (وضع) ملف في الدليل الحالي للمضيف البعيد.

يوجد عادةً تناظر (واحد لواحد) بين الأمر الذي يصدره المُستخدِم وأمر FTP الذي يُرسَل عبر توصيلة التحكم. ويتبع كل أمر يُرسَل من الزبون إلى الخادم رد يُرسَل من الخادم إلى الزبون. يتكون هذا الرد من رقم يتألف من ثلاث خانات مع رسالة اختيارية بعده. لاحظ وجه الشبه في هذا التركيب مع رمز الحالة وعبارة الحالة في سطر الحالة ضمن رسالة رد HTTP. فيما يلي أمثلة لبعض الأجوبة مع رسائلها المحتملة:

- 331 Username OK, password required

اسم المُستخدِم صحيح ومطلوب كلمة السّر.

- 125 Data connection already open; transfer starting

توصيلة البيانات مفتوحة فعلاً والنقل على وشك البدء.

- 425 Can't open data connection

تعذّر فتح توصيلة البيانات.

- 452 Error writing file

حدث خطأ عند كتابة (تخزين) الملف.

وندعو القرّاء المهتمّين بتعلم أوامر وردود FTP الأخرى للاطلاع علىRFC 959.

4-2 البريد الإلكتروني (E-mail)

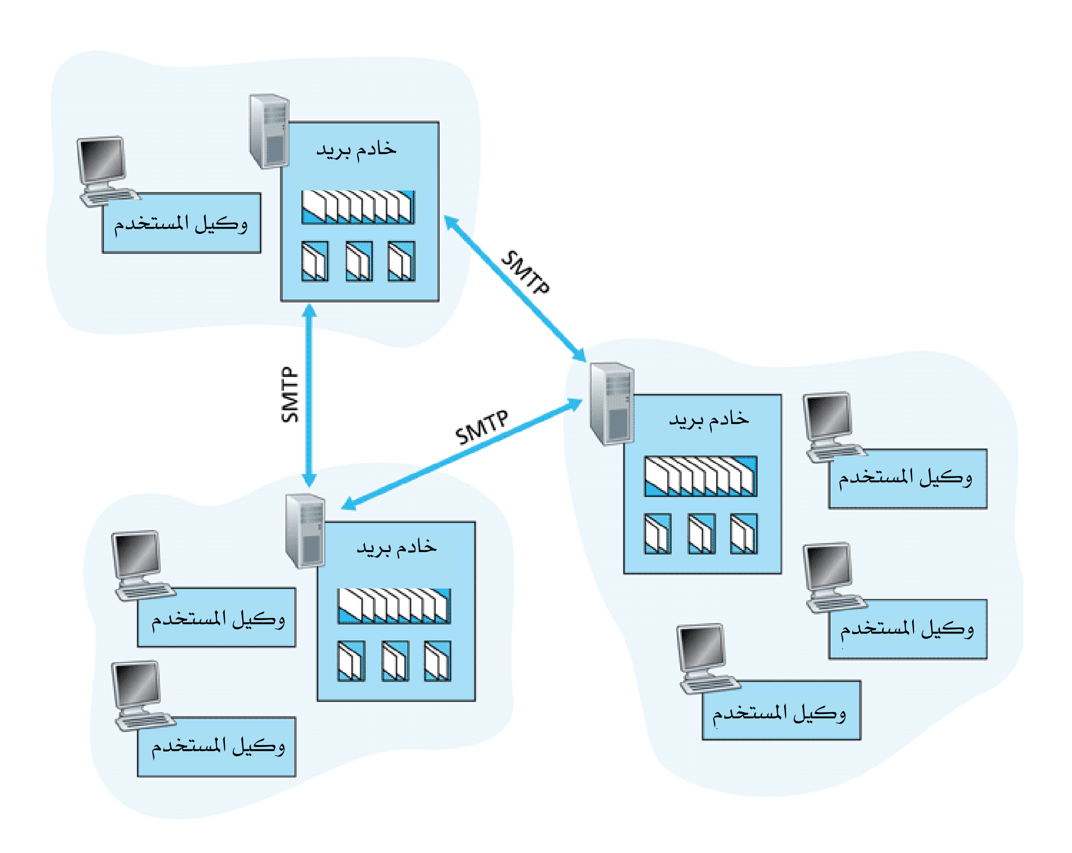
لقد ظهر البريد الإلكتروني منذ بداية الإنترنت، وكان التطبيق الأكثر شعبية عندما كانت الإنترنت في مهدها [Segaller 1998]، على مرِّ السنين أصبح هذا التطبيق أكثر إتقاناً وقوة، ويبقى اليوم أحد أكثر تطبيقات الإنترنت أهميةً وانتشاراً. كما هو الحال مع البريد العادي، يوفّرالبريد الإلكتروني وسط اتصال لاتزامني، فالناس تُرسِل وتقرأ الرسائل متى تيسر لهم ذلك بدون الحاجة للتنسيق مع جداول مواعيد الناس الآخرين. يمتاز البريد الإلكتروني مقارنةً بالبريد العادى بسرعته وسهولة توزيعه ورخص كلفته. وللبريد الإلكتروني الحديث العديد من الميزات الكبيرة كاستخدام قوائم المراسلة (العناوين) والتي عن طريقها يمكن إرسال رسائل البريد الإلكتروني ورسائل الدعاية إلى آلاف المُستخدِمين في نفس الوقت. كما تتضمن رسائل البريد الإلكتروني الحديثة الملحقات في أغلب الأحيان، والروابط التشعبية، والنصوص المنسّقة بطريقة HTML، والصور.

سنتناول في هذا الجزء بروتوكولات طبقة البرامج التي تقع في قلب بريد الإنترنت الإلكتروني. لكن قبل أن نقفز إلى مناقشة مفصّلة لتلك البروتوكولات، دعنا نلقِ نظرة مبسطة على نظام بريد الإنترنت ومكوّناته الرئيسة. يوضح الشكل 2-16 نظاماً بسيطاً لبريد الإنترنت والذي يتألف من ثلاثة مكوّنات رئيسة كما يظهر من الشكل: وكلاء المُستخدِمين، وخادمات البريد، وبروتوكول نقل البريد البسيط SMTP. نصف الآن كلاً من تلك المكوّنات ضمن سياق المُرسِل

أليس (Alice) التي ترسل رسالة بريد إلكتروني إلى المُستقبِل بوب (Bob). يسمح وكلاء المُستخدِمين لهم بقراءة الرسائل والرد عليها وإعادة توجيهها وتخزينها وتأليف رسائل جديدة. (أحياناً يُطلَق على وكلاء المُستخدِمين للبريد الإلكتروني قرّاء البريد، رغم أننا سنتفادى استخدام هذا التعبير عموماً في هذا الكتاب).

عندما تُنهي أليس إعداد رسالتها يُرسِل وكيل المُستخدِم لديها الرسالة إلى خادم البريد، حيث توضع في طابور الرسائل الخارجة. وعندما يريد بوب قراءة رسالة، يسترجع وكيل المُستخدِم لديه الرسالة من صندوق بريده في خادم البريد. في أواخر التسعينيات شاع استعمال واجهات المُستخدِم الرسومية GUI، مما سمح للمُستخدِمين بمشاهدة وتأليف رسائل الوسائط المتعددة. حالياً تعتبر برامج Microsoft Outlook، Apple Mail، Mozilla Thunderbird ضمن واجهات المُستخدِم الرسومية الشائعة للبريد الإلكتروني. كما يوجد أيضاً العديد من الواجهات النصية للبريد الإلكتروني (مثل mail، pine، elm) والمتوفرة في ساحة البرامج العامة بالإضافة إلى الواجهات المبنيّة على الويب، كما سنرى بعد قليل.

تُشكِّل خادمات البريد لب البنية التحتية للبريد الإلكتروني. لكل مُستلِم مثل بوب صندوق بريد موجود على أحد خادمات البريد تلك. ويُدير صندوق بريد بوب الرسائل التي أُرسِلت إليه ويحتفظ بها. عادةً تبدأ الرسالة رحلتها في وكيل المُستخدِم المُرسِل وتسافر إلى خادم البريد المُرسِل، ثم إلى خادم البريد المُستقبِل، حيث تُودَع في صندوق بريد المُستقبِل. وعندما يريد بوب الوصول للرسائل في صندوق بريده، يتحقق خادم البريد الذي يحتوي هذا الصندوق من شخصية بوب (عن طريق معرّف المُستخدِم وكلمة السّر). يجب أيضاً أن يتعامل خادم بريد أليس مع حالات تعطل خادم بريد بوب. إذا لم يستطع خادم أليس توصيل البريد إلى خادم بوب، يحتفظ خادم أليس بالرسالة في طابور ويحاول توصيلها لاحقاً. تُعاد المحاولة من جديد مرة كل 30 دقيقة تقريباً. وإذا لم ينجح خادم المُرسِل في ذلك بعد مُضيّ عدة أيام، يقوم الخادم بحذف الرسالة ويحيط المُرسِل عِلماً بذلك عن طريق رسالة بريد إلكتروني.



الشكل 2-16 نظرة مبسطة لنظام البريد الإلكتروني.

يعتبر SMTP بروتوكول طبقة البرامج الرئيس لبريد الإنترنت الإلكتروني. يستخدم SMTP خدمة النقل الموثوق للبيانات التي يوفرها بروتوكول TCP لنقل البريد من خادم البريد المُرسِل إلى خادم البريد المُستقبِل. وكما هو الحال مع معظم بروتوكولات طبقة البرامج، لبروتوكول SMTP جانبان: جانب الزبون (الذي يعمل على خادم البريد المُرسِل) وجانب الخادم (والذي يعمل على خادم البريد المُستقبِل). يعمل كلٌّ من جانبي الخادم والزبون لبروتوكول SMTP على كل خادم بريد،

وعندما يرسل خادم البريد رسالة إلى خادمات البريد الأخرى، فإنه يتعامل معها كزبون SMTP، وعندما يستلم خادم البريد رسالة من خادمات البريد الأخرى فإنه يتعامل معها كخادم SMTP.

2-4-1 بروتوكول نقل البريد البسيط (SMTP)

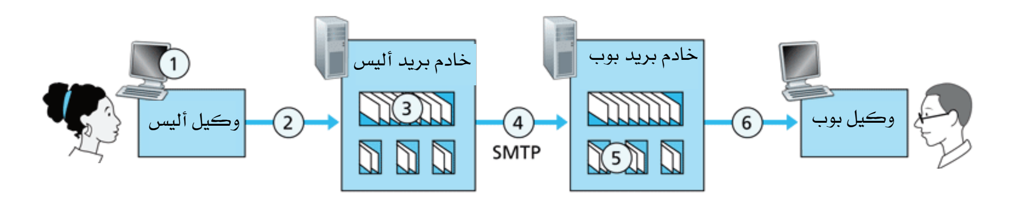
يُشكِّل بروتوكول SMTP والمُعرّف في طلب التعليقات RFC 2821 قلب بريد الإنترنت الإلكتروني. وكما ذكرنا سابقاً ينقل SMTP الرسائل من خادمات البريد المُرسِلة إلى خادمات البريد المُستقبِلة. ويعتبر SMTP أقدم بكثير من HTTP (يعود RFC الأصلي لـ SMTP إلى عام 1982 وكان SMTP موجوداً قبل ذلك بفترة طويلة). بالرغم من العديد من الميزات الرائعة لبروتوكول SMTP، والتي أدت إلى انتشاره المطلق على الإنترنت، فإنه يعتبر تقنية تراثية تعاني من بعض القيود القديمة. على سبيل المثال يتطلب البروتوكول استخدام صيغة ASCII البسيطة بطول 7 بتات لكتابة الرسالة كلها وليس فقط سطور الترويسة. كان هذا التقييد منطقياً في أوائل الثمانينيات عندما كانت قدرة الإرسال ضئيلة ولم يكن أحد يرسل مع البريد الإلكتروني ملحقات أو صوراً كبيرة، أو تسجيلات صوتية، أو ملفات فيديو. أما اليوم في عصر الوسائط المتعددة يُشكِّل استخدام ASCII بطول 7 بتات بعض المتاعب، حيث يتطلب تكويد بيانات الوسائط المتعددة الثنائية (binary) بصيغة ASCII قبل أن ترسل على SMTP؛ كما يتطلب الأمر استعادة الرسالة الثنائية من رسالة ASCII الواصلة بعد نقلها بواسطة SMTP بتكويد معاكس. تذكّر من الجزء 2-2 أن HTTP لا يتطلب تكويد بيانات الوسائط المتعددة بنمط ASCII قبل نقلها.

لتوضيح طريقة عمل SMTP الأساسية دعنا نتتبع سيناريو شائعاً. افترض أن أليس تريد أن تُرسِل رسالة ASCII بسيطة إلى بوب، ومن ثَمَّ:

1. تستدعى أليس وكيل المُستخدِم للبريد الإلكتروني لديها وتعطي عنوان البريد الإلكتروني لبوب (على سبيل المثال bob@someschool.edu) ثم تُعِد الرسالة وتعطي تعليماتها للوكيل لإرسال الرسالة.

2. يُرسِل وكيل المُستخدِم لدى أليس الرسالة إلى خادم البريد، حيث توضع في طابور الرسائل.
3. يرى جانب زبون SMTP (الذي يعمل على خادم بريد أليس) الرسالةَ في طابور الرسائل، فيفتح توصيلة TCP إلى خادم SMTP الذي يعمل على خادم بريد بوب.
4. بعد الانتهاء من خطوات المصافحة الأولية يُرسل زبون SMTP رسالة أليس إلى توصيلة TCP.
5. يستلم جانب خادم SMTP على خادم بريد بوب الرسالة، ثم يضعها في صندوق بريد بوب.
6. يستدعي بوب وكيل المُستخدِم لديه لقراءة الرسالة عندما يرغب في ذلك.

يوضح الشكل 2-17 ملخصاً لهذا السيناريو.



الشكل 2-17 إرسال رسالة بريد إلكتروني من أليس إلى بوب.

من المهم ملاحظة أن SMTP لا يستخدم عادةً خادمات البريد الوسيطة لإرسال البريد، حتى عندما يقع خادما البريد في طرفين متقابلين من العالم. فإذا كان خادم أليس في هونج كونج وخادم بوب في سانت لويس، فإن توصيلة TCP تُشكِّل توصيلةً مباشرةً بين هونج كونج وخادمات سانت لويس. وعلى وجه الخصوص إذا تعطل خادم بريد بوب تبقى الرسالة في خادم بريد أليس بانتظار محاولة إرسال جديدة ولا توضع الرسالة في بعض خادمات البريد الوسيطة.

دعنا الآن نُلقي نظرةً أكثر تفحصاً على كيفية نقل SMTP لرسالة من خادم البريد المُرسِل إلى خادم البريد المُستقبِل. سوف نلاحظ الشبه الكبير بين بروتوكول SMTP والبروتوكولات التي تُستعمَل للتفاعل الإنساني وجهاً لوجه. أولاً: يطلب زبون SMTP (والذي يعمل على مضيف خادم بريد الإرسال) من TCP إنشاء توصيلة إلى المنفذ رقم 25 في خادم SMTP (الذي يعمل على مضيف خادم بريد الاستقبال). وإذا كان خادم الاستلام المطلوب لا يعمل، يعاود الزبون المحاولة مرة أخرى لاحقاً. بمجرد إنشاء تلك التوصيلة يقوم الخادم والزبون بخطوات المصافحة (handshaking) المطلوبة في طبقة التطبيقات. تماماً كما يفعل البشر غالباً عند تقديم أنفسهم قبل تبادل المعلومات من واحد إلى آخر، يقدّم زبائن وخادمات SMTP أنفسهم قبل تبادل المعلومات. أثناء مرحلة المصافحة تلك يُشير زبون SMTP إلى عنوان البريد الإلكتروني للمُرسِل (الشخص الذي أنشأ الرسالة) وعنوان البريد الإلكتروني للمُستقبِل. وبمجرد تقديم زبون وخادم SMTP أنفسهما إلى بعضهما البعض يُرسِل الزبون الرسالة. يمكن أن يعتمد SMTP على خدمة نقل البيانات الموثوق لبروتوكول TCP لتوصيل الرسالة إلى الخادم بدون أخطاء. ثم يكرر الزبون تلك العملية على نفس توصيلة TCP إذا كان لديه رسائل أخرى يود إرسالها إلى الخادم؛ وإلا فإنه يأمر TCP بإغلاق التوصيلة.

دعنا نتناول مثالاً للرسائل المتبادلة بين زبون SMTP (C) وخادم SMTP (S). اسم مضيف الزبون هو crepes.fr واسم مضيف الخادم هو hamburger.edu. السطور التي تبدأ بـ C هي بالضبط السطور التي يرسلها الزبون إلى مقبس TCP والسطور التي تبدأ بـ S هي بالضبط السطور التي يرسلها الخادم إلى مقبس TCP. يبدأ الحوار التالي بمجرد إنشاء توصيلة TCP:

```
S: 220 hamburger.edu
C: HELO crepes.fr
S: 250 Hello crepes.fr, pleased to meet you
C: MAIL FROM: <alice@crepes.fr>
S: 250 alice@crepes.fr ... Sender ok
C: RCPT TO: <bob@hamburger.edu>
S: 250 bob@hamburger.edu ... Recipient ok
C: DATA
S: 354 Enter mail, end with "." on a line by itself
C: Do you like ketchup?
C: How about pickles?
C: .
S: 250 Message accepted for delivery
C: QUIT
S: 221 hamburger.edu closing connection
```

في هذا المثال يرسل الزبون رسالة "Do you like ketchup? How about pickles؟" من خادم البريد crepes.fr إلى خادم البريد hamburger.edu. وكجزء من الحوار يصدر الزبون خمسة أوامر: HELO (اختصار HELLO)، و MAIL FROM، وRCPT TO، وQUIT، وهي أوامر واضحة المعنى. يرسل الزبون إلى الخادم أيضاً سطراً يتكون من نقطة واحدة تُشير إلى نهاية الرسالة. (في مفردات ASCII تنتهي كل رسالة بـ CRLF، حيث تعني CR أول السطر (Carriage Return) وتعني LF تغذية سطر جديد (Line Feed)). يُصدِر الخادم رداً لكل أمر ومع كل رد رمز رد وبعض التفسير باللغة الإنجليزية (اختياري). نضيف هنا أن SMTP يستخدم توصيلات دائمة: فإذا كان لدى خادم البريد المُرسِل عدة رسائل للإرسال إلى نفس خادم البريد المُستقبِل، فإنه يمكنه إرسالها كلها على نفس توصيلة TCP. ومع كل رسالة يبدأ الزبون العملية بـ

MAIL FROM: crepes.fr

ويحدد نهاية الرسالة بنقطة على سطر مستقل، ويصدر أمر QUIT فقط بعد الانتهاء من بث كل الرسائل.

ننصحك أن تجرب بنفسك استخدام Telnet لإجراء حوار مباشر مع خادم SMTP. ولعمل ذلك أدخل الأمر

telnet serverName 25

حيث يُمثِل serverName اسم خادم البريد المحلي. وعند قيامك بذلك فإنك ببساطة تُنشئ توصيلة TCP بين مضيفك وخادم البريد المحلي على شبكتك. بعد كتابة هذا السطر يجب أن تتلقى فوراً الرد 220 من الخادم. بعد ذلك يمكنك إصدار أوامر SMTP التالية في الأوقات الملاءمة:

HELO, MAIL FROM, RCPT TO, DATA, CRLF.CRLF, QUIT

نوصي القارئ أيضاً بحل تمرين البرمجة الثانى الموجود في نهاية هذا الفصل، وفيه سيتم بناء وكيل مُستخدِم بسيط يطبّق جانب الزبون من بروتوكول SMTP، ويمكن عن طريقه إرسال رسالة بريد إلكتروني إلى مُستقبِل اعتباطي عن طريق خادم البريد المحلي.

تاريخ حالة (Case History)

بريد الهوتميل (Hotmail)

في ديسمبر/كانون الأول عام 1995 قام صابر باتيا (Sabeer Bhatia) وجاك سميث (Jack Smith) بزيارة شركة درابر فيشر جيرفيستون (Draper Fisher Jurveston) للاستثمار في مجال الإنترنت واقترحا تطوير نظام لتوفير البريد الإلكتروني مجاني من خلال الويب. كانت الفكرة هي إعطاء حساب بريد إلكتروني مجاني لأي شخص يطلبه، وجعل الحسابات بحيث يسهل الوصول إليها عن طريق الويب. فمن خلال هذا الحساب يستطيع الشخص الموصّل بالويب (مثلاً من مكتبة المدرسة أو مركز المجتمع) أن يقرأ ويرسل رسائل البريد الإلكتروني. وعلاوة على ذلك يسمح البريد الإلكتروني من خلال الويب بقابلية حركة عظيمة إلى مشتركيه. في مقابل 15٪ من الشركة قام جيرفيستون بتمويل باتيا وسميث لتأسيس شركة جديدة أطلقوا عليها الهوتميل (Hotmail) . وتمكن باتيا وسميث مع ثلاثة موظفين دائمين آخرين وحوالي 12 إلى 14 موظفاً يعملون بدوام جزئي (ولهم نسب في أسهم الشركة) من تطوير وإطلاق الخدمة في يوليو/تموز عام 1996. خلال شهر بعد إطلاق الخدمة أصبح عدد المشتركين 100 ألف مشترك. واستمر عدد المشتركين في النمو بسرعة مع عروض إعلانات الدعاية التى تظهر أثناء قراءة البريد الإلكتروني. في ديسمبر/كانون الأول عام 1997 – أي بعد أقل من 18 شهراً من إطلاق الخدمة – أصبح عدد مشتركي الهوتميل أكثر من 12 مليون مشترك عندما اشترتها شركة مايكروسوفت مقابل 400 مليون دولار.

يُنسَب نجاح الهوتميل في أغلب الأحيان إلى "سابقة ظهوره" وإلى "تسويق فيروسي" (viral marketing) متأصّل. فقد كانت شركة الهوتميل الأولى من نوعها التي وفرّت خدمة البريد الإلكتروني من خلال الويب. بالطبع قلّدت شركات أخرى فكرة الهوتميل لكن ظل الهوتميل متقدّماً بستّة شهور عليها. يتحقق الشرط الأول لنجاح مشروع بامتلاك فكرة أصلية وتطويرها بسرعة وبسرية تامة. أما الشرط الثاني فيقال: إنّ خدمةً أو منتجاً لديه تسويق فيروسي إذا سوّق المنتج نفسه. يُعتبر البريد الإلكتروني مثالاً كلاسيكياً لخدمة ذات تسويق فيروسي حيث يرسل المُرسِل رسالة إلى واحد أو أكثر من المُستقبِلين، ومن ثم يصبح كل المُستقبِلين على علم بالخدمة. برهن الهوتميل أن تحقق هذين الشرطين يمكن أن يقود إلى تطبيق ناجع. وربما سيكون بعض الطلاب الذين يقرأون هذا الكتاب من بين رجال الأعمال الجدد الذين يتخيلون ويطوّرون خدمات للإنترنت تحقق هذين الشرطين.

2-4-2 مقارنة مع بروتوكول HTTP

دعنا الآن نعقد مقارنة سريعة بين SMTP وHTTP. يُستخدَم كلا البروتوكولين لنقل الملفات من مضيف إلى آخر: ينقل HTTP الملفات (تسمى أيضاً كائنات) من خادم الويب إلى زبون الويب (المتصفّح)، بينما ينقل SMTP الملفات (أي رسائل البريد الإلكتروني) من خادم بريد إلى خادم بريد آخر. وعند نقل تلك الملفات يستخدم كلٌّ من HTTP الدائم وSMTP توصيلات دائمة. وهكذا فللبروتوكولين خصائص مشتركة. ومع ذلك فهناك أيضاً اختلافات هامة بينهما. أولاً: يُعتبر HTTP بشكلٍ أساسي بروتوكول سحب (pull protocol) – حيث يُحمِّل شخصٌ ما المعلومات على خادم الويب ويستعمل المُستخدِمون HTTP لسحب المعلومات من الخادم حسب رغبتهم. وبشكلٍ خاص فإن توصيلة TCP يتم إنشاؤها بواسطة الجهاز الذي يريد استلام الملف. في المقابل يُعتبر SMTP بروتوكول دفع (push protocol)، حيث يدفع خادم البريد المُرسِل الملف إلى خادم البريد المُستقبِل. وبشكلٍ خاص فتوصيلة TCP يتم إنشاؤها بواسطة الجهاز الذي يريد إرسال الملف أو الرسالة.

ثمة اختلاف ثانٍ كنا قد ألمحنا إليه في وقت سابق، وهو أن SMTP يتطلب أن تكون كل الرسالة – بما في ذلك جسم الرسالة – بصيغة ASCII بطول 7 بتات. فإذا كانت الرسالة تتضمن حروفاً غير ذلك (على سبيل المثال حروفاً فرنسية ذات علامات) أو تحتوي على بيانات ثنائية (كملف صورة)، فإنه يتعين تكويد الرسالة بصيغة ASCII بطول 7 بتات، في حين لا يفرض بروتوكول HTTP مثل هذه القيود على البيانات.

هناك اختلاف ثالث هام يتعلق بكيفية مناولة وثيقة تتضمن نصوصاً وصوراً (وربما مع بعض مواد الوسائط المتعددة الأخرى). كما رأينا في الجزء 2-2 يقوم HTTP بتغليف كل كائن في رسالة رد HTTP الخاصة به، بينما يضع بريد الإنترنت كل كائنات الرسالة في رسالة واحدة، كما سنرى بتفصيل أكثر لاحقاً.

2-4-3 صيغ رسائل البريد والامتداد MIME

عندما تكتب أليس رسالة بريد عادية إلى بوب قد تضمنها معلومات إضافية في أعلى الرسالة، كعنوان بوب وعنوان أليس للبريد الراجع والتاريخ. وبنفس الطريقة فعند إرسال رسالة بريد إلكتروني من شخص لآخر، فسوف يسبق جسم الرسالة نفسها مجموعة سطور تحتوي على المعلومات الإضافية (يحتوي طلب التعليقات RFC 822 على توصيف لتلك المعلومات الإضافية). يفصل بين تلك السطور وجسم الرسالة سطر فارغ (أي CRLF). كما هو الحال مع HTTP، يحتوي كل سطر ترويسة نصاً مقروءاً يتكون من كلمة دليلية تتبعها النقطتان (:) تتبعهما قيمة. بعض الكلمات الدليلية مطلوبة وبعضها اختياري. فكل ترويسة يجب أن تتضمن سطر From وسطر To، وقد تحتوي على Subject، بالإضافة إلى سطور الترويسة الاختيارية الأخرى. من المهم ملاحظة أن هذه السطور مختلفة عن أوامر SMTP التي درسناها في الجزء 2-4-1 (رغم وجود بعض الكلمات المشتركة مثل 'From' و'To'). كانت الأوامر في ذلك الجزء تُشكِّل جزءًا من بروتوكول المصافحة لـ SMTP، في حين تُشكِّل سطور الترويسة التي ذكرناها هنا جزءاً من رسالة البريد نفسها.

وكمثال تبدو سطور الترويسة لرسالة كالتالي:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Searching for the meaning of life.
```

يلي سطور الترويسة سطر فارغ ثم جسم الرسالة (بصيغة ASCII). الآن جرّب استخدام Telenet لإرسال رسالة إلى خادم البريد تحتوي على بعض سطور الترويسة بما في ذلك Subject، ولعمل ذلك استخدم الأمر

```
telnet serverName 25
```

كما سبق ذكره في الجزء 2-4-1.

امتدادات MIME للبيانات بغير صيغة ASCII

بينما تعد سطور ترويسة الرسائل التي تم توصيفها في طلب التعليقات RFC 822 كافية لإرسال نص ASCII عادي، فإنها ليست غنية بدرجة كافية لرسائل الوسائط المتعددة (كالرسائل التي تتضمن صوراً وتسجيلات صوت وفيديو) أو لنقل صيغ النص غير ASCII (كحروف اللغات الأخرى غير الإنجليزية). لإرسال المحتويات غير نصوص ASCII يجب على وكيل المُستخدِم المُرسِل استخدام سطور ترويسة إضافية في الرسالة. تم توصيف تلك السطور الإضافية في RFC 2045 وRFC 2046، ويطلق عليها "امتدادات بريد الإنترنت متعددة الأغراض" Multipurpose Internet Mail Extensions والمعروفة بالاختصار MIME وهي امتداد لطلب التعليقات RFC 822.

من بين سطور MIME الإضافية الهامة لدعم رسائل الوسائط المتعددة السطران:

Content-Type:
Content-Transfer-Encoding:

يسمح Content-Type لوكيل المُستخدِم المُستقبِل باتخاذ الإجراء المناسب لمحتوى الرسالة. على سبيل المثال بتحديد أن جسم الرسالة يحتوي على صورة JPEG يمكن لوكيل المُستخدِم المُستقبِل توجيه جسم الرسالة إلى برنامج لاسترجاع الصور المضغوطة بطريقة JPEG. ولفهم الحاجة إلى السطر الثانى Content-Transfer-Encoding تذكّر أن رسائل النصوص غير ASCII يجب أن تكوّد بصيغة ASCII لكي يفهمها SMTP، يُنبِّه هذا السطر وكيل المُستخدِم المُستقبِل إلى أن جسم الرسالة يتكون من صيغة ASCII مُكوّدة ويشير إلى طريقة التكويد المُستخدَمة. وهكذا فعندما يستلم وكيل المُستخدِم رسالة تتضمن سطري الترويسة هذين، فإنه يستخدم أولاً قيمة السطر Content-Transfer-Encoding في تحويل جسم الرسالة إلى الشكل الأصلي غير ASCII لها، وبعد ذلك يستخدم السطر Content-Type لتحديد الإجراءات المطلوب القيام بها على جسم الرسالة.

لنلقِ نظرة على مثال محدد. افترض أن أليس تريد إرسال صورة JPEG إلى بوب. ولعمل ذلك تستدعى أليس وكيل المُستخدِم للبريد الإلكتروني لديها، وتحدد عنوان البريد الإلكتروني لبوب، وتحدد موضوع الرسالة، وتدخل الصورة إلى جسم الرسالة. (وعلى حسب وكيل المُستخدِم الذي تستعمله أليس، يمكن أن تدخل الصورة إلى الرسالة كملف مرفق). وعندما تنتهي أليس من إعداد رسالتها، تنقر على "Send" (أي "أرسل")، وعندها يقوم وكيل المُستخدِم لدى أليس بتوليد رسالة MIME يمكن أن تبدو كالتالي:

```
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .......
.........................................
.......base64 encoded data
```

نلاحظ من رسالة MIME تلك أن وكيل أليس قام بتكويد صورة JPEG بطريقة base64، وهي أحد أساليب التكويد الموصوفة في MIME [RFC 2045] للتحويل إلى صيغة ASCII بطول 7 بتات والمقبولة لدى بروتوكولSMTP . هناك أسلوب تكويد آخر شائع الاستعمال هو quoted-printable، والذي يستخدم لتحويل رسالة بصيغة ASCII بطول 8 بتات (قد تتضمن حروفاً غير إنجليزية) إلى ASCII بطول 7 بتات.

عندما يقرأ بوب بريده باستعمال وكيل المُستخدِم لديه يقوم وكيل بوب بملاحظة سطر الترويسة

Content-Transfer-Encoding: base64

ومن ثم يشرع في فك التكويد لجسم الرسالة. كما تتضمن الرسالة أيضاً سطر الترويسة

Content-Type: image/jpeg

والذي يُنبِّه وكيل المُستخدِم إلى أن جسم الرسالة يجب أن يكون JPEG بعد إزالة عملية ضغط البيانات (decompression). وأخيراً تتضمن الرسالة السطر

MIME-Version: 1.0

والذي يُشير بالطبع إلى رقم إصدر MIME المُستخدَم. لاحظ أن الرسالة فيما عدا ذلك تتبع معيار RFC 822 لصيغة الرسالة. وبالتحديد بعد سطور ترويسة الرسالة يأتي سطر فارغ يتبعه بعد ذلك جسم الرسالة.

الرسائل المُستلَمة

سنكون مقصرين إذا أغفلنا ذكر نوع آخر من سطور الترويسة التي يتم إدخالها من قِبَل خادم SMTP المُستقبِل. فور استلام ذلك الخادم رسالة تحتوي على سطور ترويسة من RFC 822 وMIME يقوم الخادم بإلحاق السطر Received في بداية الرسالة. يحدد هذا السطر اسم خادم SMTP الذي أرسل الرسالة (from)، واسم خادم SMTP الذي استلم الرسالة (by)، ووقت استلام الخادم المُستقبِل للرسالة. وهكذا فإن الرسالة التي يتلقاها المُستخدِم عند الوجهة النهائية تأخذ الشكل التالي:

```
Received: from crepes.fr by hamburger.edu; 12 Oct 98 15:27:39 GMT
From: alice@crepes.fr
To: bob@hamburger.edu
Subject: Picture of yummy crepe.
MIME-Version: 1.0
Content-Transfer-Encoding: base64
Content-Type: image/jpeg

base64 encoded data .......
.......................................
.......base64 encoded data
```

كل من استعمل البريد الإلكتروني تقريباً قد رأى سطر العنوان Received يسبق الرسالة (مع سطور الترويسة الأخرى). (وهذا السطر يرى في أغلب الأحيان مباشرة على الشاشة أو عندما تُرسَل الرسالة إلى طابعة). وربما لاحظت أن رسالة واحدة قد تتضمن أحياناً عدة سطور Received، وذلك لأن الرسالة قد تُرسَل إلى أكثر من خادم SMTP في المسار بين المُرسِل والمُستلِم. على سبيل المثال إذا طلب بوب من خادم بريده الإلكتروني hamburger.edu توجيه كل رسائله إلى sushi.jp، فإن الرسالة التي يقرؤها وكيل المُستخدِم لدى بوب قد تبدأ بشيء مثل:

Received: from hamburger.edu by sushi.jp; 3 Jul 01 15:30.01 GMT
Received: from crepes.fr by hamburger.edu; 3 Jul 01 15:17:39 GMT

وتوفر سطور الترويسة تلك لوكيل المُستخدِم المُستقبِل أثراً لتتبع خادمات SMTP التي زارتها الرسالة في طريقها للوجهة، بالإضافة إلى الوقت الذي تمت فيه تلك الزيارات.

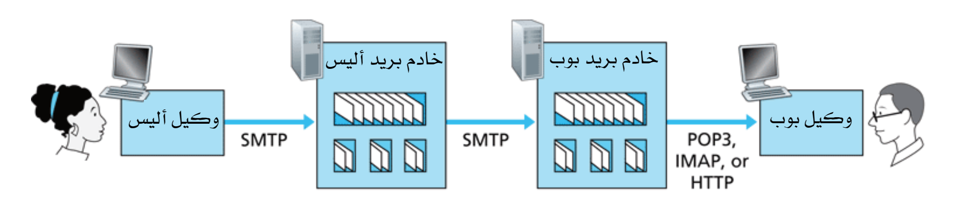
2-4-4 بروتوكولات التوصل لرسائل البريد

عندما يوصل SMTP الرسالة من خادم بريد أليس إلى خادم بريد بوب توضع الرسالة في صندوق بريد بوب. خلال هذه المناقشة افترضنا ضمنياً أن بوب يقرأ بريده بالدخول إلى مضيف الخادم ثم يشغل قارئ البريد على ذلك المضيف. حتى أوائل التسعينيات كانت تلك هي الطريقَة القياسية لعمل ذلك، أمّا اليوم فيتبع التوصل للبريد الإلكتروني البنية المعمارية من طراز زبون/خادم – حيث يقرأ المُستخدِم العادي البريد الإلكتروني باستخدام زبون يُنفَّذ على النظام الطرفي لديه، كحاسب شخصي مكتبي، أو حاسب نقّال، أو مساعد رقمي شخصي. وبتنفيذ زبون البريد على الحاسب الشخصي المحلي يتمتع المُستخدِمون بحزمة غنية من الميزات، منها القدرة على مشاهدة رسائل وملحقات الوسائط المتعددة.

افترض أن بوب (المُستقبِل) يُشغِّل وكيل المُستخدِم على حاسبه الشخصي، ومن الطبيعي اعتبار وجود خادم البريد على نفس الحاسب أيضاً. وبهذه النظرة فإن خادم بريد أليس يدير حواراً مباشراً مع حاسب بوب. ولكن هناك مشكلة

تكتنف هذه الطريقة. تذكّر أن خادم البريد يدير صناديق البريد ويُشغِّل جانبى خادم وزبون من بروتوكول SMTP. فإذا كان خادم بريد بوب يستقر على حاسبه الشخصي، يجب أن يبقى ذلك الحاسب دائماً في وضع التشغيل ومتصلاً بالإنترنت لكي يستلم البريد الجديد الذي يمكن أن يصل في أي وقت، وهذا غير عملي للعديد من مُستخدِمي الإنترنت. وبدلاً من ذلك يُشغِّل المُستخدِم عادةً وكيل المُستخدِم فقط على حاسبه الشخصي، لكنه يخزّن صندوق بريده على خادم البريد المشترك والذي يبقى دائماً في وضع تشغيل. ويدار خادم البريد المشترك هذا، والذي يشترك في استعماله المُستخدِمون الآخرون، عادةً من قِبَل موفّر خدمات الإنترنت للمُستخدِم (مثلاً جامعة أو شركة).

دعنا الآن نأخذ في الاعتبار المسار الذي تسلكه رسالة بريد إلكتروني عندما تُرسَل من أليس إلى بوب. عرفنا للتو أنه في وقتٍ ما على طول المسار من الضّروري إيداع رسالة البريد الإلكتروني في خادم بريد بوب. هذا يمكن أن يحدث ببساطة بجعل وكيل المُستخدِم أليس يُرسِل الرسالة مباشرة إلى خادم بريد بوب. ويمكن أن يحدث ذلك باستخدام SMTP (في الحقيقة صُمِّم SMTP لدفع البريد الإلكتروني من مضيف إلى آخر). ومع ذلك فإنه في العادة لا يدير وكيل المُستخدِم المُرسِل حواراً مباشراً مع خادم البريد المُستلم. بدلاً من ذلك وكما هو موضح في الشكل 2-18 يستخدم وكيل المُستخدِم لدى أليس بروتوكول SMTP لدفع رسالة البريد الإلكتروني إلى خادم البريد لديها، ثم يستخدم خادم بريد أليس بروتوكول SMTP (كزبون SMTP) لترحيل رسالة البريد الإلكتروني إلى خادم بريد بوب. لماذا إجراء خطوتين؟ بالدرجة الأولى لأنه بدون الترحيل من خلال خادم بريد أليس ليس لوكيل المُستخدِم لدى أليس أي إمكانية للوصول إلى خادم بريد الوجهة النهائية إذا كان من المتعذر الوصول إليه في الوقت الحالي. وبجعل أليس تُودِع البريد الإلكتروني أولاً في خادم بريدها الخاص، يمكن أن يعاود خادم بريد أليس محاولة إرسال الرسالة إلى خادم بريد بوب مراراً وتكراراً (مثلاً كل 30 دقيقة) لحين أن يصبح خادم بريد بوب شغّالاً. يحدد طلب التعليقات الخاص ببروتوكول SMTP كيفية استخدام أوامر SMTP لترحيل رسالة عبر عدة خادمات SMTP.



الشكل 2-18 بروتوكولات البريد الإلكتروني وكياناتها المتصلة.

لكن ما زالت هناك قطعة واحدة مفقودة من اللغز! كيف يحصل مُستقبِل مثل بوب يُشغّل وكيل المُستخدِم على حاسبه الشخصي على رسائله الموجودة في خادم البريد عند موفّر خدمة الإنترنت له؟ لاحظ أن وكيل المُستخدِم لدى بوب لا يستطيع استخدام SMTP للحصول على تلك الرسائل، لأن الحصول على الرسائل عملية سحب بينما SMTP بروتوكول دفع. يكتمل اللغز بتوفير نظام خاص للوصول للبريد بنقل الرسائل من خادم بريد بوب إلى حاسبه الشخصي. يوجد حالياً عدد من بروتوكولات الوصول للبريد شائعة الاستخدام تشمل الإصدار الثالث لبروتوكول مكتب البريد (Post Office Protocol--Version 3 (POP3))، وبروتوكول الوصول لبريد الإنترنت (Internet Mail Access Protocol (IMAP))، وبروتوكول HTTP.

يلخص الشكل 2-18 البروتوكولات المستخدمة مع بريد الإنترنت: يُستخدَم SMTP لنقل البريد من خادم البريد المُرسِل إلى خادم البريد المُستقبِل؛ ويُستخدَم SMTP أيضاً لنقل البريد من وكيل المُستخدِم المُرسِل إلى خادم البريد المُرسِل. ويُستخدَم بروتوكول الوصول للبريد مثل POP3 لنقل البريد من خادم البريد المُستقبِل إلى وكيل المُستخدِم المُستقبِل.

بروتوكول POP3

يُعتبر POP3 بروتوكولاً بسيطاً جداً للوصول للبريد، وهو مُعرَّف في طلب التعليقات RFC 1939 (وهو مستند قصير وسهل القراءة). ولأن البروتوكول بسيط

جداً فوظيفته أيضاً محدودة. يبدأ POP3 العمل عندما يفتح وكيل المُستخدِم (الزبون) توصيلة TCP إلى خادم البريد (الخادم) على منفذ رقم 110. بعد ذلك يمر POP3 بثلاث مراحل: التحقق (authorization)، وتنفيذ العملية (transaction)، والتحديث (update). في المرحلة الأولى (التحقق) يُرسِل وكيل المُستخدِم اسم المُستخدِم وكلمة السّر (بشكلٍ مقروء) للتحقق من شخصية المُستخدِم والترخيص له. وأثناء المرحلة الثانية (تنفيذ العملية) يسترجع وكيل المُستخدِم الرسائل؛ يمكن لوكيل المُستخدِم أثناء هذه المرحلة أيضاً أن يُؤشِّر على رسائل للحذف، أو يلغي علامات الحذف، أو يحصل على إحصائيات عن بريده. المرحلة الثالثة (التحديث) وتحدث بعد أن يُصدِر الزبون الأمر QUIT ليُنهِي جلسةَ POP3، ومنها يحذف خادم البريد الرسائل التي أُشِّرت للحذف.

في أثناء المرحلة الثانية (تنفيذ العملية) يُصدِر وكيل المُستخدِم الأوامر ويرد الخادم على كل أمر. هناك إجابتان محتملتان: "+OK" (يتبعها أحياناً بيانات من الخادم إلى الزبون) ويستخدمها الخادم للإشارة إلى سلامة الأمر السابق على ما يرام؛ و"-ERR" ويستخدمها الخادم للإشارة إلى حدوث خطأ عند تنفيذ الأمر السابق.

تتضمن مرحلة التحقق أمرين رئيسين:

```
user <username>
pass <password>
```

ونقترح أن تستخدم Telnet للاتصال مباشرة مع خادم POP3 على منفذ رقم 110، وأن تجري الحوار التالي على افترض أن mailServer هو اسم خادم البريد:

```
telnet mailServer 110
+OK POP3 server ready
user bob
+OK
pass hungry
+OK user successfully logged on
```

إذا أخطأت في كتابة أمر فسيجيب خادم POP3 برسالة خطأ "-ERR".

الآن لنلقِ نظرة على مرحلة تنفيذ العملية. في أغلب الأحيان يمكن تجهيز وكيل المُستخدِم الذي يستعمل POP3 لكي "يُنزّل ويحذف" (download and delete) أو "يُنزّل ويحتفظ" (download and keep). تعتمد سلسلة الأوامر التي تصدرمن قِبَل وكيل مُستخدِم POP3 على نمط التشغيل المستخدم من بين هذين النمطين. في النمط "download and delete" سوف يُصدِر وكيل المُستخدِم الأوامر:

list, retr, dele

وكمثال افترض أن المُستخدِم لديه رسالتان في صندوق بريده وأجرى الحوار التالي، حيث ترمز C لوكيل المُستخدِم (الزبون) وS لخادم البريد (الخادم):

```
C: list
S: 1 498
S: 2 912
S:.
C: retr 1
S: (blah blah ...
S:  ................
S: ................ blah)
S:.
C: dele 1
C: retr 2
S: (blah blah ...
S:  ................
S: ..........blah)
S:.
C: dele 2
C: quit
S: +OK POP3 server signing off
```

في هذا الحوار يطلب وكيل المُستخدِم أولاً من خادم البريد عرض حجم كلٍّ من الرسائل المُخزَّنة، ثم يسترجع ويحذف كل رسالة من الخادم. لاحظ أنه بعد مرحلة التوثيق يستعمل وكيل المُستخدِم أربعة أوامر فقط:

list, retr, dele, quit

وصيغة هذه الأوامر مُعرَّفة في طلب التعليقات RFC 1939. بعد تنفيذ الأمر quit يدخل خادم POP3 مرحلة التحديث ويحذف الرسائل رقم 1 و2 من صندوق البريد.

من مشاكل نمط "download-and-delete" هذا أن المُستقبِل بوب قد يكون متجولاً ويريد الوصول إلى رسائل بريده من عدة أجهزة، على سبيل المثال من حاسب مكتبه وحاسب بيته وحاسبه النقّال. هذا النمط يقسم رسائل بريد بوب على هذه الأجهزة الثلاثة. وبالتحديد إذا قرأ بوب رسالة على حاسب مكتبه فلن يكون قادراً على إعادة قراءتها من حاسبه النقّال في البيت في وقت لاحق. أما في نمط "download-and-keep" فيترك وكيل المُستخدِم الرسائل على خادم البريد بعد تحميلها. في هذه الحالة يمكن أن يعيد بوب قراءة الرسائل من الأجهزة المختلفة؛ ويمكن أن يستعرض رسالة في مكتبه ثم يستعرضها ثانيةً في وقت لاحق من البيت.

يحتفظ خادم POP3 ببعض المعلومات عن الحالة أثناء نفس الجلسة بين وكيل المُستخدِم وخادم البريد؛ وبصفة خاصة يتتبع أي رسائل للمُستخدِم قد أُشّرت للحذف. لكن لا يحتفظ خادم POP3 بمعلومات عن الحالة من جلسة لأخرى مما يبسط كثيراً عملية تطوير برنامج الخادم.

بروتوكول IMAP

في بروتوكول POP3 للوصول للبريد يمكن أن يُنشئ بوب مجلدات للبريد (mail folders) على حاسبه الشخصي ويخزن رسائله فيها. كما يمكنه بعد ذلك حذف الرسائل أو نقلها بين المجلدات أو البحث عن رسائل بعينها (باسم المُرسِل أو الموضوع). لكن هذا الأسلوب (أي وجود المجلدات والرسائل في الحاسب المحلي) يُشكِّل مشكلة للمُستخدِم المتنقل الذي يُفضِّل إبقاء المجلدات والرسائل على الخادم البعيد ليتمكن من الوصول إليها من أي حاسب آخر. هذا الأمر غير ممكن مع بروتوكول POP3، فهو لا يوفر أية وسيلة للمُستخدِم لإنشاء المجلدات البعيدة وتخزين الرسائل فيها.

لحل هذه المشكلة وغيرها من المشاكل، تم استحداث بروتوكول IMAP والمُعرَّف في RFC 3501، وهو بروتوكول للوصول للبريد مثل POP3، غير أنه يتميز عنه بميزاتٍ عديدة ولكنه أيضاً أكثر تعقيداً. يقرن خادم IMAP كل رسالة بمجلد، وعندما تصل رسالة إلى الخادم لأول مرة ترتبط بمجلد صندوق الوارد (INBOX). عندئذ يمكن للمُستخدِم نقل الرسالة إلى مجلد جديد تم إنشاؤه من قبل، أو قراءة الرسالة، أو حذفها، وهكذا. يوفر بروتوكول IMAP أوامر تسمح للمُستخدِمين بإنشاء المجلدات ونقل الرسائل من مجلد لآخر. كما يوفر الأوامر التي تُمكِّن المُستخدِمين من البحث في المجلدات البعيدة عن رسائل تطابق معايير معينة. لاحظ أنه بخلاف POP3 يحتفظ خادم IMAP بمعلومات عن حالة المُستخدِم عبر جلسات IMAP كأسماء المجلدات والرسائل بكلٍ منها.

من الميزات الهامة الأخرى لبروتوكول IMAP أنه يتضمن الأوامر التي تسمح لوكيل المُستخدِم بالحصول على مكوّنات الرسائل. على سبيل المثال يمكن أن يحصل وكيل المُستخدِم على سطور الترويسة فقط من الرسالة أو على جزء واحد فقط من رسالة MIME متعددة الأجزاء. هذه الميزة مفيدة عندما تكون وصلة الاتصال ذات سعة إرسال منخفضة بين وكيل المُستخدِم وخادم البريد (على سبيل المثال وصلة مودم بطيئة السرعة). في هذه الحالة قد لا يريد المُستخدِم تنزيل كل الرسائل في صندوق بريده ليتفادى الرسائل الطويلة خصوصاً التي قد تحتوي مثلاً على تسجيلات صوتية أو لقطات فيديو. يمكن أن تقرأ كل شيء عن IMAP من على موقعه الرسمي على الويب [IMAP 2007].

البريد الإلكتروني من خلال الويب (Web-Based E-mail)

اليوم يرسل العديد من المُستخدِمين بشكلٍ متزايد رسائل البريد الإلكتروني ويستعرضونه من خلال متصفّحات الويب. قدّم الهوتميل (Hotmail) خدمة بريد الويب في منتصف التسعينيات، كما يتوفر البريد الإلكتروني على الويب الآن أيضاً من خلال ياهوو (Yahoo) وجوجل (Google) بالإضافة إلى كل جامعة وشركة كبرى تقريباً. وبهذه الخدمة يكون وكيل المُستخدِم متصفّح ويب

عادي، ويتصل المُستخدِم بصندوق بريده البعيد عن طريق HTTP. وعندما يريد مُستخدِم مثل بوب الوصول لرسالة في صندوق بريده، تُرسَل الرسالة من خادم بريد بوب إلى متصفّح بوب بواسطة HTTP بدلاً من POP3 أو IMAP. عندما يريد مُرسِل مثل أليس إرسال رسالة بريد إلكتروني، تُرسَل الرسالة من متصفّحها إلى خادم البريد على HTTP بدلاً من SMTP. ومع ذلك فإن خادم بريد أليس لا يزال يُرسِل رسائل إلى خادمات البريد الأخرى ويتلقّى الرسائل منها باستخدام SMTP.

5-2 خدمة دليل الإنترنت لأسماء النطاقات (DNS)

يمكن أن نُميِّز – نحن البشر – بعدة طرق. على سبيل المثال يمكن أن نُميِّز بالأسماء في شهادات الميلاد، أو بأرقام الضمان الاجتماعي، أو أرقام رخص القيادة. ورغم أن كل أشكال التعريف هذه يمكن أن تُستعمَل لتمييز الناس، قد يكون أحد الطرق أكثر ملاءمة من الآخر ضمن سياق معين. على سبيل المثال تفضل الحاسبات في وكالة IRS (وكالة تحصيل الضرائب المشهورة في الولايات المتّحدة) استخدام أرقام الضمان الاجتماعي ثابتة الطول بدلاً من أسماء شهادات الميلاد، في حين يُفضِّل الناس استخدام أسماء شهادات الميلاد لسهولة تذكرها بدلاً من أرقام الضمان الاجتماعي.

ومثلما يمكننا تمييز البشر بعدة طرق، يمكننا أن تُميِّز مضيفات الإنترنت بعدة طرق. أحد طرق التعريف هو اسم المضيف (hostname) مثل: cnn.com، www.yahoo.com، cis.poly.edu، gala.cs.umass.edu. وهي سهلة التذكر ولذا يفضلها الناس. ومع ذلك فإن أسماء المضيفات تعطي معلومات قليلة عن مواقع المضيفات على شبكة الإنترنت. مثلاً اسم المضيف www.eurecom.fr والذي ينتهي برمز البلد fr، يخبرنا بأنه من المحتمل أن يكون المضيف في فرنسا، لكن لا يمكننا القول أكثر من ذلك. وعلاوة على ذلك يمكن أن تكون أسماء المضيفات بأطوال مختلفة (حروف وأرقام) مما يُصعِّب معالجتها بالموجّهات. لهذه الأسباب تُميَّز المضيفات أيضاً بما يسمّى عناوين IP.

سوف نناقش عناوين IP ببعض التفصيل في الفصل الرابع، ولكن من المفيد تناولها ببضع كلمات قصيرة الآن. يتكون عنوان IP من أربعة بايتات، وله تركيب هرمي ثابت. يأخذ عنوان IP شكلاً كالمثال التالي: 121.7.106.83، حيث يفصل كل بايت عن الآخر بنقطة ويكتب بطريقة عشرية من 0 إلى 255. عنوان IP ذو تركيب هرمي لأنه عندما نقرأ العنوان من اليسار إلى اليمين، نحصل على معلومات معينة أكثر فأكثر حول مكان المضيف في الإنترنت (أي ضمن أية شبكة في شبكة الشبكات العالمية). بنفس الطريقة عندما نقرأ العنوان البريدي من أسفل إلى أعلى نحصل على معلومات معينة أكثر فأكثر حول مكان المُرسَل إليه.

1-5-2 الخدمات المتوفرة من قِبَل DNS

رأينا للتو أن هناك طريقتين لتمييز المضيف: اسم المضيف وعنوان IP. يفضل الناس اسم المضيف لسهولة تذكره، بينما تفضل أجهزة الموجّهات عناوين IP ثابتة الطول وذات التركيب الهرمي. ولكي نوفق بين هذه التفضيلات المختلفة، نحتاج إلى خدمة الدليل التي تترجم أسماء المضيفات إلى عناوين IP، وتلك هي المهمة الرئيسة لنظام أسماء النطاقات في الإنترنت ((Domain Name System (DNS)). فنظام DNS هو (1) قاعدة بيانات موزّعة مُنفّذة في خادمات DNS ذات ترتيب هرمي، (2) أحد بروتوكولات طبقة التطبيقات والذي يسمح للمضيفات بالبحث في قاعدة البيانات الموزّعة تلك. غالباً ما تعمل خادمات DNS على أجهزة يونيكس بنظام ((Berkeley Internet Name Domain (BIND)). ويستخدم بروتوكول DNS بروتوكول UDP على منفذ 53.

في الغالب يُستخدَم DNS عن طريق بروتوكولات طبقة البرامج الأخرى بما فيها HTTP، وSMTP، وFTP لترجمة أسماء المضيفات إلى عناوين IP. كمثال لنأخذ في الاعتبار ما يحدث عندما يقوم متصفّح يعمل على مضيف ما بطلب صفحة الويب:

www.someschool.edu/index.html

لكي يتمكن المضيف من إرسال رسالة طلب HTTP إلى خادم الويب www.someschool.edu يجب أن يحصل أولاً على عنوان IP لهذا الخادم. ويتم ذلك كالتالي:

1. يشغّل نفس جهاز المُستخدِم برنامج الزبون لتطبيق DNS.
2. يقتبس المتصفّح اسم المضيف www.someschool.edu من عنوان URL ويرسله إلى برنامج الزبون لتطبيق DNS.
3. يرسل برنامج زبون DNS استفساراً يحتوي على اسم المضيف إلى خادم DNS.
4. يستقبل زبون DNS في النهاية إجابة تتضمن عنوان IP لاسم المضيف.
5. عندما يستلم المتصفّح عنوان IP من DNS، يمكنه أن يُنشئ توصيلة TCP مع عملية خادم HTTP والموجودة على منفذ رقم 80 في ذلك العنوان.

نرى من هذا المثال أن DNS يتسبب في زمن تأخير إضافي – كبير أحياناً – لتطبيقات الإنترنت التي تستخدمه. لحسن الحظ – كما سنناقش فيما بعد – يُحفَظ عنوان IP المطلوب في أغلب الأحيان في الذاكرة المخبأة لخادم DNS قريب، مما يساعد في تخفيض حركة مرور بيانات DNS على الشبكة وكذلك تخفيض زمن تأخير DNS في المتوسط.

يوفر DNS بضع خدمات أخرى مهمة بالإضافة لترجمة أسماء المضيفات إلى عناوين IP:

- أسماء المضيف البديلة (Host aliases): يمكن أن يحمل مضيف له اسم صعب واحداً أو أكثر من الأسماء البديلة. على سبيل المثال يمكن أن يأخذ المضيف relayl.west-coast.enterprise.com اسمين أكثر شهرة مثل

enterprise.com
www.enterprise.com

في هذه الحالة يقال إن اسم المضيف

relayl.west-coast.enterprise.com

هو الاسم القانوني (أو الرسمي) (canonical name). وتُعتبر أسماء المضيفات البديلة أكثر سهولة للتذكر من أسماء المضيفات القانونية. ويمكن أن يستخدم تطبيق DNS للحصول على اسم المضيف القانوني وعنوان IP المناظرين لعنوان بديل للمضيف.

- أسماء خادم البريد البديلة (Mail server aliasing): لأسباب واضحة من المرغوب فيه جداً أن تكون عناوين البريد الإلكتروني سهلة التذكر. على سبيل المثال إذا كان بوب لديه حساب هوتميل، فإن عنوان بريده الإلكتروني قد يكون بسيطاً مثل bob@hotmail.com. ومع ذلك فإن اسم مضيف خادم البريد الهوتميل أكثر تعقيداً وأصعب في تذكره من الاسم البسيط hotmail.com (فقد يكون اسم المضيف القانوني مثلاً relayl.west-coast.hotmail.com). يمكن أن يستخدم DNS من قبل تطبيق بريدي للحصول على اسم المضيف القانوني لاسم مضيف بديل بالإضافة إلى عنوان IP للمضيف. في الحقيقة يسمح سجل MX (كما سترى فيما بعد) أن يكون لخادم البريد وخادم الويب لشركةٍ ما أسماء مضيفات بديلة متطابقة. على سبيل المثال يمكن أن يسمى كلٌّ من خادم الويب وخادم البريد للشركة enterprise.com.
- توازن الأحمال (Load balancing): يُستخدَم DNS أيضاً للقيام بتوزيع الأحمال بين الخادمات المكررة بشكلٍ متوازن، كخادمات الويب المكررة (replicated web servers). يتم تكرار خدمات الويب للمواقع ذات الأحمال العالية (التي يكثر عليها الطلب) مثل enn.com على خادمات متعددة، ويعمل كلٌّ منها على نظام طرفي مستقل وله عنوان IP مختلف. ولهذه الخادمات مجموعة من عناوين IP المرتبطة باسم مضيف قانوني واحد. تحتوي قاعدة بيانات DNS على مجموعة عناوين IP تلك. وعندما ترسل زبائن DNS استفساراً عن اسم مرتبط بتلك المجموعة من العناوين، يرد الخادم بمجموعة عناوين IP كاملة ولكن بترتيب مختلف للعناوين في كل مرة. لأن الزبون عادة ما يرسل رسالة طلب HTTP إلى

عنوان IP المدرج أولاً في المجموعة، فإن إعادة ترتيب DNS للعناوين يُوزِّع حركة البيانات بين الخادمات المكررة. يستخدم أيضاً تدوير DNS للعناوين في البريد الإلكتروني لكي يتسنى استخدام خادمات بريد متعددة لنفس الاسم البديل. ومؤخراً استخدمت شركات توزيع المحتوى (مثل [Akamai 2007]) بروتوكول DNS بطرق أكثر تطوراً للقيام بتوزيع محتوى الويب (انظر الفصل السابع).

تم توصيف بروتوكول DNS في RFC 1034 وRFC 1035 وعُدِّل في عدة طلبات تعليقات أخرى. سنتناول هنا باختصار السّمات الرئيسة فقط لطريقة عمله، ونحيل القارئ المهتم لبعض المصادر الإضافية كبعض طلبات التعليقات الأخرى والكتاب [Abitz 1993] والأبحاث [Mockapetris 1988; Mockapetris 2005].

المبادئ في الواقع العملي (Principles in Practice)

بروتوكول DNS وأداء وظائف حرجة للشبكة عن طريق بنية زبون/خادم

يعتبر بروتوكول DNS أحد بروتوكولات طبقة التطبيقات مثله في ذلك مثل HTTP وFTP وSMTP، وذلك للأسباب التالية: (1) أنه يستخدم بين أنظمة طرفية تتصل فيما بينها باستخدام أسلوب زبون/خادم، و(2) أنه يعتمد على بروتوكولات النقل التحتية لتبادل رسائل DNS بين الأنظمة الطرفية المتصلة. ولكن من وجهة نظر أخرى يختلف دور DNS تماماً عن الويب، ونقل الملفات، وتطبيقات البريد الإلكتروني، فهو ليس تطبيقاً يتعامل معه المُستخدِم مباشرة، ولكن DNS يوفر خدمة رئيسية للإنترنت – ألا وهي خدمة ترجمة أسماء المضيفات إلى عناوين IP المناظرة لتطبيقات المُستخدِم والبرامج الأخرى في الإنترنت. لاحظنا في الجزء 1-2 أن معظم التعقيد في البنية المعمارية للإنترنت يوجد على "حواف" الشبكة. ويعتبر DNS الذي يقوم بعملية ترجمة الاسم إلى العنوان والذي يستخدم زبائن وخادمات موجودة على حافة الشبكة مثالاً آخر لفلسفة التصميم تلك.

2-5-2 نظرة عامة على كيفية عمل DNS

لنلقِ الآن نظرة عامة مبسطة على كيفية عمل DNS. ستُركِّز مناقشتنا على خدمة ترجمة اسم المضيف إلى عنوان IP. افترض أن تطبيقاً ما (مثل متصفّح الويب أو قارئ بريد)، يجري تشغيله على مضيف يحتاج لترجمة اسم مضيف إلى عنوان IP، سوف يستدعي التطبيق جانب الزبون لـ DNS ويحدد اسم المضيف الذي يحتاج لترجمته. (على العديد من أجهزة يونيكس، يُستخدم الإجراء gethostbyname() للحصول على الترجمة. سنوضح في الجزء 2-7 كيفية استدعاء DNS من تطبيقات جافا). عندئذ يبدأ DNS العمل في مضيف المُستخدِم بإرسال رسالة استفسار إلى الشبكة. ترسل كل رسائل DNS للاستفسار والرد ضمن وحدة بيانات UDP إلى منفذ رقم 53. بعد زمن تأخير يتراوح من عدة ميللي ثانية إلى عدة ثوانٍ، يتلقى DNS في مضيف المُستخدِم رسالة رد DNS تتضمن المطابقة المطلوبة بين الاسم والعنوان. ترسل تلك المطابقة إلى التطبيق الطالب. وهكذا – من منظور التطبيق الطالب في مضيف المُستخدِم – يبدو DNS كصندوق أسود يوفر خدمة ترجمة مباشرة وبسيطة. ولكن في حقيقة الأمر هذا الصندوق الأسود الذي يوفر تلك الخدمة معقّد، ويتكون من عدد كبير من خادمات DNS الموزّعة حول العالم، بالإضافة إلى بروتوكول في طبقة التطبيقات يحدد كيف تتصل خادمات DNS بالمضيفات المستفسرة.

وكتصميم بسيط لـ DNS يُستخدم خادم DNS واحد يحتوي على كل المطابقات بين الأسماء والعناوين. في هذا التصميم المركزي يوجّه الزبائن كل الاستفسارات إلى ذلك الخادم الذي يتولى الردّ مباشرة عليها. على الرغم من بساطة هذا التصميم وجاذبيته، فهو للأسف غير ملائم للإنترنت اليوم لاتساعها وتزايد عدد المضيفات. ومن بين المشاكل الأخرى لهذا التصميم المركزي:

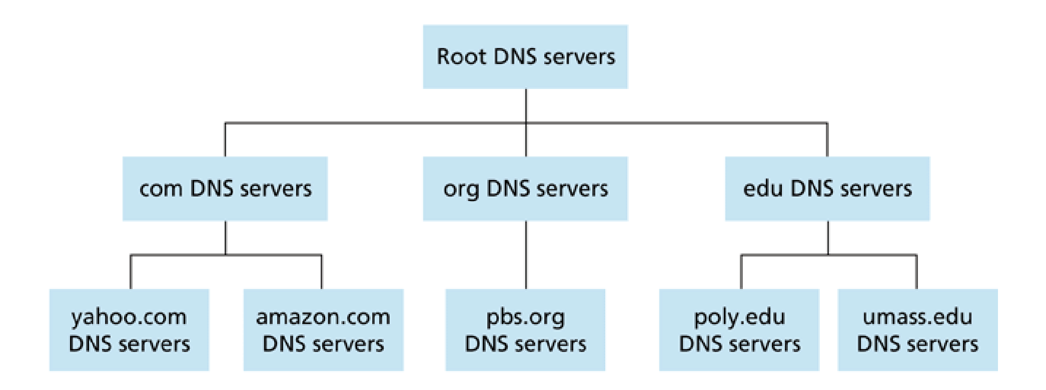
- نقطة وحيدة للعطل: إذا تعطل خادم DNS فسوف تتعطل أيضاً الإنترنت بالكامل!

- زيادة حجم حركة المرور: يجب أن يعالج خادم DNS الوحيد كل استفسارات DNS (لكل طلبات HTTP ورسائل البريد الإلكتروني الناشئة من مئات الملايين من المضيفات).
- قاعدة البيانات مركزية وبعيدة: لا يمكن أن يكون خادم DNS الوحيد "قريباً من" كل الزبائن المستفسرة. فإذا وضعنا خادم DNS الوحيد في مدينة نيويورك، فعلى كل الاستفسارات من استراليا أن تسافر إلى الجانب الآخر من المعمورة، وربما على وصلات اتصال بطيئة ومزدحمة، مما يؤدي إلى زيادة في التأخير.
- الصيانة: يجب أن يحتفظ خادم DNS الوحيد بالسجلات لكل مضيفات الإنترنت. ليست المشكلة فقط في أن قاعدة البيانات المركزية هذه ستكون ضخمة، ولكن سيتعين أيضاً تحديثها باستمرار لتشمل كل مضيف جديد.

وباختصار فإن قاعدة البيانات المركزية في خادم DNS وحيد لا تتناسب ببساطة مع التوسع الكبير في حجم الشبكة. وفي الواقع يعتبر DNS مثالاً رائعاً لكيفية تحقيق قاعدة بيانات موزّعة على الإنترنت.

قاعدة بيانات هرمية وموزّعة

لكي يتعامل DNS مع قضية التوسع المضطرد في الإنترنت، فإنه يستخدم عدداً كبيراً من الخادمات منظّمة بترتيب هرمي وموزّعة حول العالم. في هذا التصميم لا يحتوي خادم DNS وحيد على كل المطابقات لكل المضيفات على الإنترنت وإنما توجد موزّعة عبر عدة خادمات. كتبسيط أولي توجد ثلاثة أصناف من تلك الخادمات مرتبة بشكلٍ هرمي (كما هو مبين في الشكل 2-19): خادمات DNS الجذرية (Root DNS Servers)، خادمات DNS لنطاق المستوى الأعلى ((Top-Level Domain (TLD))، وخادمات DNS المسؤولة (Authoritative DNS Servers).



الشكل 2-19 جزء من التوزيع الهرمي لخادمات DNS.

ولفهم كيفية تعامل هذه الأصناف الثلاثة من الخادمات مع بعضها، افترض أن زبون DNS يريد تحديد عنوان IP لاسم المضيف www.amazon.com في أول تبسيط تتم الأحداث التالية: يتصل الزبون بأحد خادمات الجذر أولاً، والتي ترجع عناوين IP لخادمات TLD لنطاق المستوى الأعلى com. ثم يتصل الزبون بأحد خادمات TLD تلك، والتي ترجع عنوان IP لخادم مسؤول عن amazon.com. أخيراً يتصل الزبون بأحد الخادمات المسؤولة عن amazon.com والذي يرجع عنوان IP لاسم المضيف www.amazon.com. سوف نفحص لاحقاً عملية البحث هذه لدى DNS (DNS lookup) بتفصيل أكثر، ولكن دعنا أولاً نلقي نظرة أدق على تلك الأصناف الثلاثة من خادمات DNS:

- خادمات DNS الجذرية: يوجد على الإنترنت 13 خادم DNS جذري (مسماة من A إلى M)، أغلبها في أمريكا الشمالية. يوضح الشكل 2-20 التوزيع الجغرافي لها في أكتوبر/تشرين عام 2007. وتوجد قائمة بهذه الخادمات في [Root-servers 2007]. وبالرغم من أننا أشرنا إلى كلٍّ منها كما لو كان خادماً وحيداً، فإن كل "خادم" في الحقيقة عبارة عن مجموعة (cluster) من الخادمات المكررة من أجل زيادة الاعتمادية (reliability) والأمن (security).



الشكل 2-20 التوزيع الجغرافي لخادمات DNS الجذرية في عام 2007.

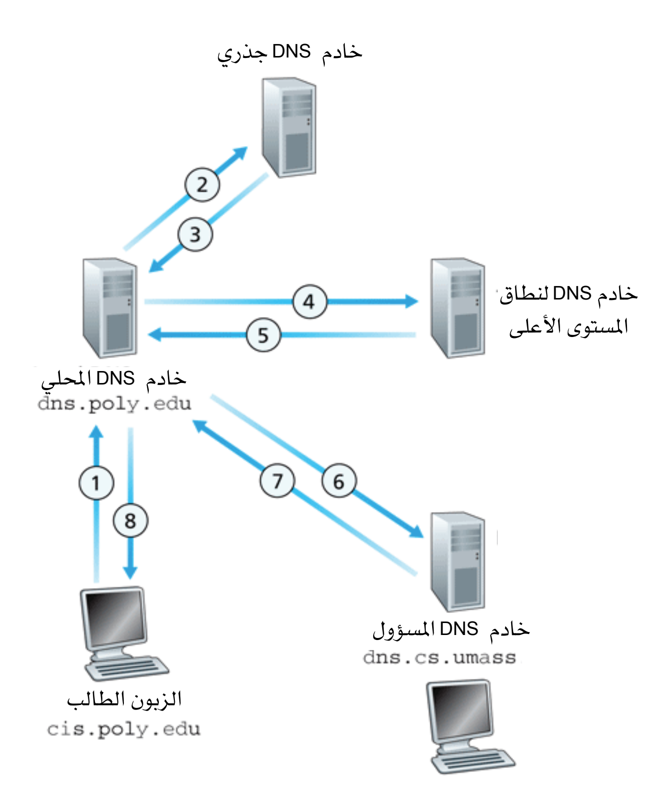
- خادمات نطاق المستوى الأعلى (TLD): هذه الخادمات مسؤولة عن نطاقات المستوى الأعلى مثل com، org، net، edu، gov وكذلك كل نطاقات المستوى الأعلى التي تمثل البلدان مثل uk، fr، ca، jp. وفي وقت كتابة هذا الكتاب (ربيع عام 2007) كانت شركة حلول الشبكات (Network Solutions) مسؤولة عن خادمات TLD لنطاق المستوى الأعلى com، وشركة Educause عن خادمات TLD لنطاق المستوى الأعلى edu.
- خادمات DNS المسؤولة: يتعين على كل منظمة لها مضيفات متاحة للوصول العام على الإنترنت (مثل خادمات الويب وخادمات البريد) توفير سجلات DNS سهلة الوصول بشكلٍ عام لتحويل أسماء تلك المضيفات إلى عناوين IP، ويحتفظ خادم DNS المسؤول الخاص بتلك المنظمة بسجلات DNS تلك. يمكن أن تختار المنظمة أن توفر بنفسها خادم DNS المسؤول الخاص بها لحفظ تلك السجلات. وكبديل لذلك يمكن أن تدفع المنظمة رسوماً لبعض موفّري الخدمة (service providers) مقابل تخزين السجلات على خادم DNS مسؤول لديها. معظم الجامعات والشركات

الكبيرة تحتفظ بخادمات DNS مسؤولة أساسية وثانوية (احتياطية) خاصة بها.

تُنظَم كل خادمات DNS (الجذرية وTLD والمسؤولة) في تركيب هرمي (شجري) كما هو مبين في الشكل 2-19. هناك نوع آخر مهم من خادمات DNS يطلق عليه خادمات DNS المحلية. وللدقة لا ينتمى خادم DNS المحلي لشجرة الخادمات، ولكنه مع ذلك يلعب دوراً محورياً في بنية DNS المعمارية. يوجد لدى كل موفّر خدمة إنترنت (كجامعة، أو قسم أكاديمي، أو شركة، أو موفّر خدمة الإنترنت السكني) خادم DNS محلي (يسمى خادم الاسم الاعتيادي (default name server)). عندما يتصل مضيف مع موفّر خدمة إنترنت فإن موفّر الخدمة يعطي المضيف عنوان IP لواحد أو أكثر من خادمات DNS المحلية لديه (عادةً من خلال بروتوكول DHCP والذي سنتناوله في الفصل الرابع). يمكنك معرفة عنوان IP لخادم DNS المحلي لديك بسهولة بفتح نوافذ حالة الشبكة في نظام تشغيل ويندوز أو يونيكس. ويكون خادم DNS المحلي عادة "قريباً من" المضيف. في حالة موفّر خدمة إنترنت لمؤسّسة قد يكون خادم DNS المحلي على نفس شبكة الاتصالات المحلية LAN كالمضيف. أما في حالة موفّر خدمة الإنترنت السكني فعادة ما يفصل بينه وبين المضيف ما لا يزيد عن بضعة موجّهات (routers). وعندما يُرسِل مضيف استفسار DNS، يُرسَل الاستفسار إلى خادم DNS المحلي، والذي يتصرف كـ "وكيل" (proxy) فيُرسِل الاستفسار إلى شجرة خادمات DNS كما سنناقش بتفصيل أكثرفيما بعد.

لنلقِ نظرة على مثال بسيط. افترض أن المضيف cis.poly.edu يرغب في الحصول على عنوان IP للمضيف gaia.cs.umass.edu، وافترض أيضاً أن خادم DNS المحلي لجامعة Polytechnic يُدعى dns.poly.edu وأن خادم DNS المسؤول عن gaia.cs.umass.edu يُدعى dns.umass.edu. كما هو موضح في الشكل 2-21 يرسل المضيف cis.poly.edu أولاً رسالة استفسار DNS إلى خادم DNS المحلي dns.poly.edu تتضمن اسم المضيف الذي يريد ترجمته (أي gaia.cs.umass.edu). يوجه خادم DNS المحلي رسالة الاستفسار إلى خادم DNS

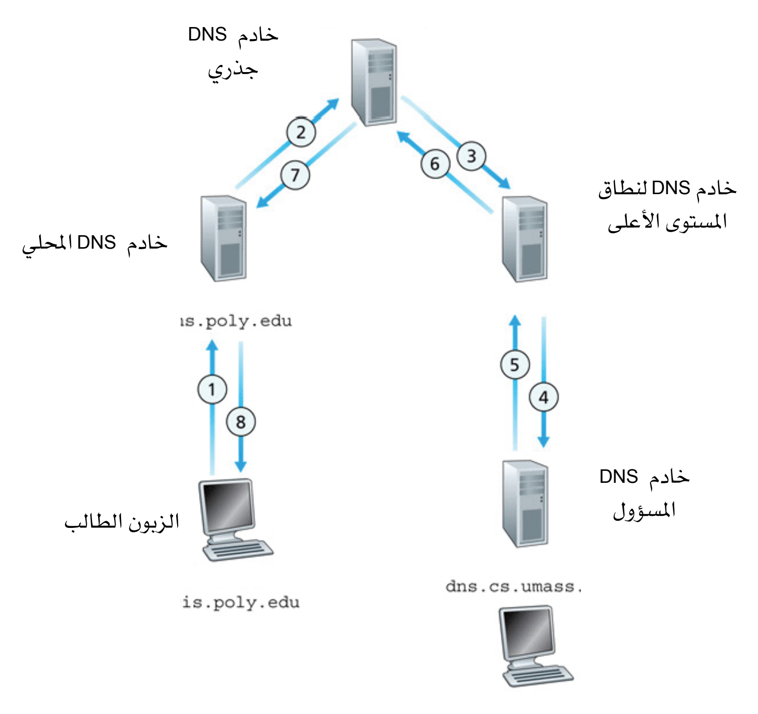
الجذري والذي يلاحظ اللاحقة edu ويرجع إلى خادم DNS المحلي قائمة عناوين IP لخادمات TLD المسؤولة عن نطاق edu. بعد ذلك يرسل خادم DNS المحلي رسالة الاستفسار مرة ثانية إلى أحد خادمات TLD تلك، والذي يلاحظ اللاحقة umass.edu، فيرد بعنوان IP لخادم DNS المسؤول لجامعة ماسوشوستس (أي dns.umass.edu). أخيراً يُرسِل خادم DNS المحلي رسالة الاستفسار مرة ثانية مباشرةً إلى dns.umass.edu والذي يرد بعنوان IP لـ gaia.cs.umass.edu. لاحظ في هذا المثال أنه لكي نحصل على العنوان المرتبط باسم مضيف واحد، تم إرسال ثماني رسائل DNS: أربع رسائل استفسار وأربع رسائل رد! وسنرى قريباً كيف يستخدم DNS الذاكرة المخبأة ليُخفّض حركة المرور الناجمة عن تلك الاستفسارات.



الشكل 2-21 التفاعل بين خادمات DNS المختلفة.

في مثالنا السابق افترضنا أن خادم TLD يعرف خادم DNS المسؤول الذي يتضمن اسم المضيف، ولكن ذلك لن يكون صحيحاً دائماً. بدلاً من ذلك فإن خادم TLD قد يعرف فقط خادم DNS وسيط والذي يعرف بدوره خادم DNS المسؤول الذي يتضمن اسم المضيف. على سبيل المثال افترض ثانية أن جامعة ماسوشوستس لها خادم DNS يسمى dns.umass.edu وافترض أيضاً أن كل قسم من أقسام الجامعة له خادم DNS خاص به وأن ذلك الخادم مسؤول عن كل المضيفات في القسم. في هذه الحالة عندما يستلم خادم DNS الوسيط dns.umass.edu استفساراً عن مضيف ينتهي اسمه بـ cs.umass.edu فسوف يرجع إلى dns.poly.edu عنوان IP لـ dns.cs.umass.edu والذي هو مسؤول عن كل أسماء المضيفات التي تنتهي بـ cs.umass.edu ثم يُرسِل خادم DNS المحلي dns.poly.edu الاستفسار إلى خادم DNS المسؤول الذي يُرجِع العنوان المطلوب إلى خادم DNS المحلي والذي يُرجِع بدوره العنوان إلى المضيف الطالب. في هذه الحالة تُرسَل عشر رسائل DNS!

يستخدم المثال الموضح في الشكل 2-21 كلاً من الاستفسارات التتابعية (recursive) والاستفسارات التكرارية (iterative). الاستفسار الذي أُرسِل من المضيف cis.poly.edu إلى الخادم dns.poly.edu استفسار تتابعي، وفيه يطلب المضيف من الخادم أن يحصل له على العنوان نيابة عنه. لكن الاستفسارات الثلاثة التالية تكرارية لأن كل الردود عليها تعود مباشرةً إلى dns.poly.edu. نظرياً يمكن أن يكون أي استفسار DNS تتابعي أو تكراري. على سبيل المثال يوضح الشكل 2-22 سلسلة استفسارات DNS كلها تتابعية. عملياً عادة ما تتبع الاستفسارات النمط الموضح في الشكل 2-21، أي يكون الاستفسار من المضيف الطالب إلى خادم DNS المحلي تتابعياً، بينما بقية الاستفسارات تكون تكرارية.



الشكل 2-22 استفسارات DNS التتابعية.

ذاكرة DNS المخبأة

أهملنا في مناقشتنا حتى الآن استخدام DNS للذاكرة المخبأة والتي تمثل ميزة هامة جداً لنظام DNS. في الحقيقة يستخدم DNS الذاكرة المخبأة على نطاق واسع لكي يُحسّن الأداء بتقليل زمن التأخير وتخفيض عدد رسائل DNS التي تجوب الإنترنت. الفكرة وراء استخدام DNS للذاكرة المخبأة بسيطة للغاية. ففي سلسلة استفسارات عندما يستلم خادم DNS رد DNS (يتضمن مثلاً مطابقة بين اسم مضيف وعنوان IP)، يمكن أن يحتفظ بنسخة منه في ذاكرته المحلية. على سبيل المثال في الشكل 2-21 في كل مرة يتلقى خادم DNS المحلي dns.poly.edu رداً من خادم DNS يمكن أن يُخزّن أياً من المعلومات الواردة في الرد في تلك الذاكرة.

إذا وُجد زوج البيانات المكون من اسم المضيف وعنوان IP المناظر مخزناً على خادم DNS ووصل استفسار جديد إلى الخادم لنفس اسم المضيف، يمكن أن يرد الخادم بعنوان IP المطلوب، حتى إذ لم يكن هو الخادم المسؤول عن اسم المضيف. ولأن المضيفات والمطابقة بين أسمائها وعناوين IP المقابلة لها ليست ثابتة على الإطلاق (نظراً لتخصيص العناوين ديناميكياً) فإن خادمات DNS تحذف المعلومات المخبأة بعد فترة معينة (تحدد غالباً بيومين).

وكمثال افترض أن مضيف apricot.poly.edu يريد أن يسأل dns.poly.edu عن عنوان IP للمضيف cnn.com. علاوة على ذلك افترض أنه بعد ساعات قليلة يقوم مضيف آخر من جامعة بولي تكنك مثلاً kiwi.poly.edu بالاستفسار من dns.poly.edu عن اسم المضيف نفسه. مع استخدام الذاكرة المخبأة، سيكون خادم DNS المحلي قادراً على إرجاع عنوان IP لـ cnn.com فوراً للمضيف الثانى المستفسر بدون الحاجة للاستفسار من أي من خادمات DNS الأخرى. ويمكن أيضاً لخادم DNS المحلي أن يُخزّن عناوين IP لخادمات TLD مما يسمح لخادم DNS المحلي بتخطى خادمات DNS الجذرية في سلسلة استفسار (وهذا ما يحدث في أغلب الأحيان).

3-5-2 سجلات ورسائل DNS

تقوم خادمات DNS (والتي تكوّن فيما بينها قاعدة بيانات DNS الموزّعة) بتخزين سجلات الموارد (Resource Records (RRs)) بما في ذلك السجلات الخاصة بالمطابقة ما بين اسم المضيف وعنوان IP المناظر. تحمل كل رسالة رد DNS سجلاً واحداً أو أكثر. سنلقي في هذا الجزء والجزء الذي يليه نظرة عامة وسريعة عن سجلات ورسائل DNS، ويمكن الحصول على تفاصيل أكثر من [Abitz 1993] أو من طلبات التعليقات حول DNS [RFC 1034; RFC 1035].

يتكون سجل المورد من الأربعة حقول التالية:

(Name, Value, Type, TTL)

يُمثل الحقل TTL فترة العمر لسجل المورد، وهو يحدد متى يجب أن يحذف السجل من الذاكرة المخبأة. في الأمثلة التي نقدّمها فيما يلي أهملنا حقل TTL. ويعتمد معنى الحقل Name والحقل Value على قيمة حقل Type كما يلي:

- إذا كانت قيمة Type = A ، فعندئذ تمثل قيمة الحقل Name اسم المضيف، بينما تمثل قيمة الحقل Value عنوان IP المناظر لذلك الاسم. وهكذا يعطي السجل من نوع A المطابقة القياسية بين اسم المضيف وعنوان IP المناظر. وكمثال لهذا النوع من السجلات:

(relayl.bar.foo.com, 145.37.93.126, A)

- إذا كانت قيمة Type = NS فإن قيمة الحقل Name تمثل اسم نطاق (مثل foo.com) وقيمة الحقل Value تمثل اسم خادم DNS المسؤول عن هذا النطاق والذي يعرف كيف يحصل على عناوين IP للمضيفات في ذلك النطاق. يُستخدَم هذا السجل لإعادة توجيه استفسارات DNS في سلسلة الاستفسار. وكمثال لهذا النوع:

(foo.com, dns.foo.com, NS)

- إذا كانت قيمة Type = CNAME فإن قيمة الحقل Name تمثل اسم المضيف القانوني لاسم بديل للمضيف والتي تذكر في حقل Value. يمكن أن يعطي هذا السجل رداً للاستفسار عن الاسم القانوني لاسم مضيف ما. وكمثال لهذا النوع:

(foo.com, relayl.bar.foo.com, CNAME)

- إذا كانت قيمة حقل Type = MX فإن قيمة حقل Name تمثل الاسم القانوني لخادم البريد للاسم البديل للمضيف الموجود في حقل Value. وكمثال لهذا النوع:

(foo.com, mail.bar.foo.com, MX)

- تسمح سجلات MX بإعطاء أسماء شهرة بديلة بسيطة لأسماء مضيفات خادمات البريد. لاحظ أنه باستعمال السجل MX يمكن أن تستخدم

شركة نفس الاسم البديل لخادم البريد ولأحد خادماتها الأخرى (مثل خادم الويب لديها). وللحصول على الاسم القانوني لخادم البريد سوف يستفسر زبون DNS عن سجل MX، وللحصول على الاسم القانوني للخادم الآخر سوف يستفسر زبون DNS عن سجل CNAME.

إذا كان خادم DNS هو الخادم المسؤول عن اسم مضيف معين فإن خادم DNS سيتضمن سجلاً من النوع A لاسم المضيف. (حتى إذا كان خادم DNS ليس الخادم المسؤول فقد يتضمن أيضاً سجلاً من النوع A في ذاكرته المخبأة). إذا كان الخادم ليس مسؤولاً عن اسم المضيف فإنه سيتضمن سجلاً من النوع NS للنطاق الذي يتضمن اسم المضيف، وكذلك أيضاً سجلاً من النوع A يوفر عنوان IP لخادم DNS الموجود في حقل Value لسجل NS. كمثال افترض أن خادم TLD مسؤول عن النطاق edu ولكنّه ليس مسؤولاً عن المضيف gaia.cs.umass.edu عندئذ سيتضمن هذا الخادم سجلاً للنطاق الذي يضم المضيف cs.umass.edu على سبيل المثال:

(umass.edu, dns.umass.edu, NS)

وسيتضمن خادم TLD الخاص بنطاق edu أيضاً سجلاً من النوع A يحدد عنوان IP المناظر لاسم خادم أسماء النطاقات dns.umass.edu على سبيل المثال:

(dns.umass.edu, 128.119.40.111, A)

رسائل DNS

أشرنا سابقاً في هذا الجزء إلى رسائل DNS الخاصة بالاستفسار والرد، وهذان هما النوعان الوحيدان من أنواع رسائل DNS. كما أن كلاً من رسائل الاستفسار والرد لهما نفس الصيغة كما هو موضح في الشكل 2-23.

تتلخص معاني الحقول المختلفة في رسالة DNS فيما يلي:

يمثل أول 12 بايتاً الجزء الخاص بسطور الترويسة، والذي يتضمن بدوره عدداً من الحقول. الحقل الأول هو رقم تعريفي (identifier) يميز

الاستفسار ويتكون من 16 بتاً. ينسخ هذا المعرّف في رسالة الرد على ذلك الاستفسار، مما يسمح للزبون بربط الردود التي يتلقاها بالاستفسارات التي أرسلها. يوجد عدد من الأعلام (flags) في حقل الأعلام. يشير أحد الأعلام إلى نوع الرسالة (0 يعني رسالة استفسار، و1 يعني رسالة رد)، ويشير آخر في رسالة الرد إلى أن الخادم هو الخادم المسؤول عن اسم المضيف الجاري الاستفسار عنه. تكون قيمة علَم البحث التتابعي (recursion) 1 عندما يرغب الزبون (مضيف أو خادم DNS) في أن يقوم خادم DNS بالبحث التتابعي في حالة عدم وجود السجل المطلوب لديه. وإذا كان الخادم يدعم البحث التتابعي فإنه يضع القيمة 1 في ذلك العلَم في رسالة الرد. تتضمن الترويسة أيضاً أربعة حقول يشير كلٌّ منها إلى طول كل قسم من الأقسام الأربعة التي تلي الترويسة.

أعلام (flags)	رقم تعريفي
عدد سجلات المورد بالأجوبة	عدد الاستفسارات
عدد سجلات المورد الإضافية	عدد سجلات المورد للخادمات المسؤولة
الاسم والنوع لحقول الاستفسارات (عدد متغير من سجلات المورد)	
سجلات المورد في رسائل الرد على الاستفسارات (عدد متغير من سجلات المورد)	
سجلات المورد الخادمات المسؤولة (عدد متغير من سجلات المورد)	
المعلومات الإضافية الأخرى المفيدة (عدد متغير من سجلات المورد)	

12بايتاً

الشكل 2-23 صيغة رسالة DNS.

- يحتوي "قسم السؤال" (question section) على معلومات عن الاستفسار المطلوب. ويتضمن هذا القسم: (1) حقل "Name" ويحتوي على الاسم المستفسر عنه، (2) حقل "Type" ويشير إلى نوع الاستفسار المطلوب، على سبيل المثال عنوان المضيف المناظر لاسم (النوع A) أو خادم البريد المناظر لاسم (النوع MX).
- في رسالة رد من خادم DNS، يحتوي "قسم الرد" (answer section) على سجلات الموارد للاسم المستفسر عنه في الأصل. تذكّر أنه يوجد في كل سجل مورد حقل النوع (على سبيل المثال: A، NS، CNAME، MX)، وحقل القيمة، وفترة العمر TTL. يمكن أن ترجع رسالة الرد العديد من سجلات المورد (RR) في الجواب، لأن اسم المضيف يمكن أن يكون له عدة عناوين IP (كما في حالة خادمات الويب المكررة كما ناقشنا سابقاً في هذا الجزء).
- يحتوي "قسم المسؤولية" (authority section) على سجلات الخادمات المسؤولة (authoritative servers) الأخرى.
- يحتوي "القسم الإضافي" (additional section) على سجلات أخرى مساعدة. على سبيل المثال يحتوي حقل الرد في رسالة الرد لاستفسار MX على سجل مورد يعطي اسم المضيف القانوني لخادم البريد. كما يحتوي القسم الإضافي على سجل من النوع A يعطي عنوان IP لاسم المضيف القانوني لخادم البريد.

كيف ترسل رسالة استفسار DNS مباشرةً من المضيف الذي تعمل عليه إلى خادم DNS؟ يمكن القيام بذلك بسهولة باستخدام برنامج البحث nslookup، والمتوفر على معظم أجهزة ويندوز ويونيكس. مثلاً من مضيف ويندوز افتح واجهة الأوامر (command prompt) واستدعِ برنامج nslookup، ويتم ذلك ببساطة بكتابة nslookup. بعد استدعاء البرنامج يمكن أن ترسل استفسار DNS إلى أي خادم DNS (جذر، أو TLD، أو مسؤول). بعد استلام رسالة الرد من خادم DNS تستعرض أداة البحث السجلات المُتضمِّنة في الرد (بشكلٍ يسهل على المُستخدِم

قراءته). وكبديل لتشغيل أداة البحث nslookup من مضيفك الخاص، يمكن أن تزور أحد مواقع الويب العديدة التي تسمح لك باستخدام أداة البحث عن بُعد. (فقط اكتب "nslookup" في أي محرك بحث على شبكة الويب وستجلب إليك أحد تلك المواقع).

إدخال سجلات إلى قاعدة بيانات DNS

ركزت المناقشة السابقة على كيفية استرجاع سجلات من قاعدة بيانات DNS، وقد تتساءل وكيف أدخلت تلك السجلات في قاعدة البيانات في المقام الأول؟ دعنا ننظر إلى كيفية تحقيق ذلك في سياق مثال معين. افترض أنك أنشأت شركة جديدة تسمى شبكة Utopia، فإن أول شيء تريده بالتأكيد هو أن تسجل اسم النطاق networkutopia.com لدى أحد المسجِّلين (registrars). والمسجِّل هو كيان تجاري يقوم بالتحقق من تفرد اسم النطاق (أي عدم استخدامه من قِبَل جهة أخرى)، وكذلك إضافة أسماء النطاقات إلى قاعدة بيانات DNS (كما سنناقش فيما بعد) وذلك نظير أجر بسيط مقابل خدماته. قبل عام 1999 كان المسجِّل الوحيد هو شركة Network Solutions، وكان لديها احتكار على تسجيل النطاقات com، net، org. ولكن الآن هناك العديد من المسجِّلين يتنافسون على الزبائن، وتقوم شركة الإنترنت للأسماء والأعداد المُخصَّصة (Internet Corporation for Assigned Names and Numbers (ICANN)) بترخيص المسجِّلين المعتمدين. توجد قائمة كاملة بالمسجِّلين المعتمدين على الموقع http://www.internic.net.

عندما تسجل اسم النطاق networkutopia.com مع مسجِّل ما ستحتاج أيضاً لتزويد المسجِّل بالأسماء وعناوين IP لخادمات DNS الأساسية والثانوية المسؤولة. افترض أن الأسماء وعناوين IP هي:

dns1.networkutopia.com
dns2.networkutopia.com
212.212.212.1
212.212.212.2

سوف يتأكد المسجِّل من أن سجلاً من نوع NS وآخر من نوع A قد أدخلت ضمن خادمات TLD للنطاق com، وذلك لكلٍّ من هذين الخادمين المسؤولين. وبالتحديد، يقوم المسجِّل بإدخال السجلين التاليين إلى نظام DNS بخصوص الخادم المسؤول الأساسي للنطاق networkutopia.com:

(networkutopia.com, dnsl.networkutopia.com, NS)
(dnsl.networkutopia.com, 212.212.212.1, A)

يجب أيضاً التأكد من إضافة سجل من النوع A لخادم الويب www.networkutopia.com وسجل من النوع MX لخادم البريد mail.networkutopia.com إلى خادمات DNS المسؤولة. لوقت قريب كانت محتويات كلٍّ من خادمات DNS تضبط بطريقة ثابتة (على سبيل المثال من ملف تهيئة يقوم بإعداده مدير النظام). ومؤخراً أُضيف خيار UPDATE إلى نظام DNS للسماح بالإضافة أو الحذف من قاعدة البيانات عن طريق رسائل DNS بطريقة ديناميكية. ويوجد توصيف التعديل الديناميكي لنظام DNS في RFC 2136 وRFC 3007.

بمجرد الانتهاء من كل تلك الخطوات، سيكون بوسع الجمهور زيارة موقع الشركة وإرسال البريد الإلكتروني إلى موظفيها. دعنا نختم مناقشتنا عن DNS بإثبات صحة هذه الجملة. يساعد هذا الإثبات أيضاً في تقوية ما تعلمناه عن DNS.

افترض أن أليس في استراليا أرادت استعراض صفحة الويب www.networkutopia.com. كما ناقشنا في وقت سابق سيرسل مضيفها أولاً استفسار DNS إلى خادم DNS المحلي. بعد ذلك سيتصل خادم DNS المحلي بخادم TLD المسؤول عن النطاق com (يجب أيضاً أن يتصل خادم DNS المحلي بخادم DNS الجذري إذا كان عنوان خادم TLD المسؤول عن النطاق com غير موجود بالذاكرة المخبأة). يحتوي خادم TLD هذا على سجل من النوع NS وآخر من النوع A كما عرضنا سابقاً، حيث تأكد المسجِّل من إدخال تلك السجلات إلى كل خادمات TLD للنطاق com . يرسل خادم TLD للنطاق com رداً لخادم DNS المحلي لدى أليس يحتوي على السجلين. ثم يرسل خادم DNS المحلي استفسار DNS إلى

212.212.212.1 يسأل عن سجل من النوع A المناظر لـ www.networkutopia.com. يعطي هذا السجل عنوان IP لخادم الويب المطلوب (افترض أنه 212.212.71.4) والذي يرسله خادم DNS المحلي إلى مضيف أليس. يمكن أن يبدأ متصفّح أليس الآن توصيلة TCP إلى المضيف 212.212.71.4 ويرسل طلب HTTP على التوصيلة. وكما ترى فإن ما يجرى خلف كواليس الإنترنت أكثر بكثير مما تراه العين عند تصفح الويب!

نبذة عن الأمن (Focus on Security)

نقاط ضعف DNS (الثغرات الأمنية في DNS)

رأينا أن DNS هو أحد المكونات الرئيسة للبنية التحتية للإنترنت حيث لا يمكن أن تعمل بدونه العديد من الخدمات الهامة مثل الويب والبريد الإلكتروني. والسؤال الطبيعي هو كيف يمكن الهجوم على DNS؟ هل يقبع DNS منتظراً أن يُبعَد عن الخدمة متسبباً في تعطل الكثير من تطبيقات الإنترنت؟

أول ما يخطر على الذهن هو هجوم فيضان الحيّز الترددي الموزّع لحجب الخدمة (Distributed Denial of Service (DDoS)) (انظر الجزء 1-6) ضد خادمات DNS. على سبيل المثال يمكن أن يحاول مهاجم إرسال سيل من الرزم إلى كل خادم من خادمات DNS الجذرية بحيث لا يستطيع الرد على العديد من استفسارات DNS الشرعية. في الحقيقة حدث مثل هذا الهجوم على نطاق واسع ضد خادمات DNS الجذرية في 21 أكتوبر/تشرين الأول عام 2002، حيث قام المهاجمون باستخدام شبكة الروبوت (botnet) لإرسال كم هائل من رسائل ICMP للبينج إلى كلٍّ من خادمات DNS الجذرية الثلاثة عشر (سنناقش رسائل ICMP في الفصل الرابع لكن يكفي الآن أن تعرف أنها أنواع خاصة من وحدات بيانات IP). لحسن الحظ تسبب هذا الهجوم الواسع النطاق في إحداث أقل ما يمكن من الأضرار والتى لم يتأثر بها الكثير من مُستخدِمي الإنترنت، فبرغم نجاح المهاجمين في توجيه سيل الرزم إلى الخادمات الجذرية إلا أن العديد منها كانت محمية بمرشحات الرزم (packet filters) والمهيئة لمنع كل رسائل ICMP للبينج الموجهة نحو خادمات الجذر. ساعد ذلك على استمرار الخادمات المحمية في العمل بشكلٍ طبيعي. علاوة على ذلك تحتفظ معظم خادمات DNS المحلية بعناوين IP لخادمات نطاق المستوى الأعلى (TLD) مما يسمح لعملية الاستفسار

بتجاوز خادمات DNS الجذرية في أغلب الأحيان.

هجوم آخر قد يكون أكثر فعّالية هو شن هجوم DDoS ضد خادمات نطاق المستوى الأعلى (TLD) بإرسال سيل من استفسارات DNS لتلك الخادمات، كإرسال كم هائل من استفسارات DNS إلى كل خادمات المستوى الأعلى للنطاق com سيكون ترشيح استفسارات DNS الموجّهة إلى خادمات DNS أكثر صعوبة وكذلك لا يمكن تجاوز تلك الخادمات بنفس السهولة التي يتم بها تجاوز خادمات DNS الجذرية. لكن يمكن الحد من شدة مثل هذا الهجوم باستخدام ذاكرة مخبأة في خادمات DNS المحلية.

من الممكن مهاجمة DNS بطرق أخرى. مثلاً في هجوم "رجل في الوسط" (man-in-the-middle) يعترض المهاجم الاستفسارات من المضيفات ويرجع ردوداً مزيفة. وفي هجوم تسميم DNS يرسل المهاجم إجابات مزيفة إلى خادم DNS والذى ينخدع ويضيف السجلات المزيفة إلى ذاكرته المخبأة. يمكن أن تستغل تلك الهجمات على سبيل المثال لتوجيه مُستخدِم ويب إلى موقع الويب للمهاجم. ومع ذلك فهذه الهجمات صعبة التطبيق حيث تتطلب القدرة على اعتراض الرزم أو خنق الخادمات [Skoudis 2006].

وهناك هجوم DNS آخر هام لكنه ليس هجوماً على خدمة DNS في حد ذاته وإنما يستغل البنية التحتية لـ DNS لشنّ هجوم DDoS ضد مضيف مستهدف (على سبيل المثال خادم البريد الإلكتروني لجامعتك). في هذا الهجوم يرسل المهاجم استفسارات DNS إلى العديد من خادمات DNS المسؤولة بكلٍّ منها عنوان مصدر مغشوش للمضيف المستهدف. حينئذ ستقوم خادمات DNS بإرسال الأجوبة مباشرة إلى المضيف المستهدف (لاحظ أن المضيف المستهدف لم يرسل أي استفسار في حقيقة الأمر). إذا أمكن تشكيل الاستفسارات بحيث يكون حجم رسائل الرد أكبر بكثير (في عدد البايتات) من الاستفسار (وهو ما يسمى بالتضخيم (amplification)) ففي هذه الحالة يمكن أن يغمر الهدف لدرجة تمنعه من توليد معظم حركة مرور بياناته الخاصة. إلا أن نجاح مثل تلك الهجمات الإنعكاسية التي تستغل DNS ما زال محدوداً حتى الآن [Mirkovic 2005].

وخلاصة القول أن DNS قد أثبت قدرته على صد مثل تلك الهجمات ضده. حتى الآن لم ينجح هجوم في عرقلة خدمة DNS، ورغم ذلك كانت هناك بعض الهجمات الإنعكاسية الناجحة ولكن أمكن التصدي لها بالإعداد المناسب لخادمات DNS.

2-6 تطبيقات النظائر (Peer-to-Peer Applications)

ذكرنا في الجزء 2-1-1 أنه بشكلٍ عام يمكن تصميم التطبيق باستخدام بنية معمارية تعتمد على مفهوم "زبون/خادم" أو باستخدام بنية معمارية تعتمد على مفهوم "النظائر". تستخدم كل التطبيقات التي وصفناها حتى الآن في هذا الفصل (بما في ذلك الويب، والبريد الإلكتروني، وخدمة أسماء النطاقات DNS) البنية المعمارية "زبون/خادم" حيث تعتمد أساساً على خادمات البنية التحتية التي تعمل دائماً (always-on). في حين أن بنية النظائر P2P – كما ذكرنا – تعتمد بأقل قدر ممكن (أو لا تعتمد على الإطلاق) على خادمات البنية التحتية التي تعمل دائماً. بدلاً من ذلك تتصل مباشرةً أزواج من المضيفات (تسمى النظائر) بشكلٍ متقطع مع بعضها البعض. وتلك النظائر ليست ملكاً لموفّر الخدمة وإنما هي حاسبات مكاتب وحاسبات نقالة تحت سيطرة المُستخدِمين.

وسوف نتناول في هذا الجزء ثلاثة تطبيقات مختلفة تلائم تماماً تصميم بنية النظائر. التطبيق الأول هو توزيع الملفات، وفيه يتم توزيع ملف من مصدر وحيد إلى عدد كبير من النظائر. ويعتبر توزيع الملفات تطبيقاً ملائماً لبدء دراسة بنية النظائر، حيث يُظهِر بشكلٍ واضح القدرة الذاتية على التوسع (self-scalability) لتلك البنية. وكمثال محدد لتوزيع الملفات سوف نصف نظام BitTorrent الشائع. أما تطبيق النظائر الثاني الذي سنتناوله فهو تنظيم المعلومات والبحث عنها في مجتمع من النظائر، حيث سنستكشف عدة أنظمة مختلفة يطبق كلٌّ منها في نظم مشاركة النظائر للملفات على نطاق واسع (large-scale file sharing). في التطبيق الثالث والأخير سنستكشف سكايب (Skype)، وهو تطبيق نظائر ناجح بشكلٍ هائل لهاتف الإنترنت.

2-6-1 توزيع النظائر للملفات

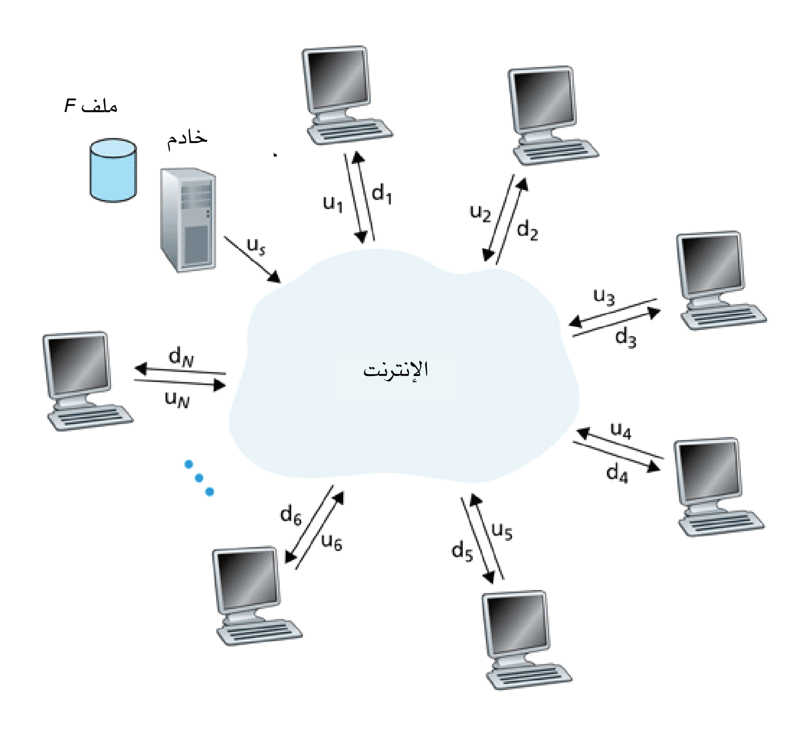
لنبدأ رحلتنا مع بنية النظائر بدراسة تطبيق طبيعي جداً، ألا وهو توزيع ملف كبير الحجم من خادم وحيد إلى عدد كبير من المضيفات (النظائر). قد يكون

هذا الملف نسخة جديدة من نظام تشغيل لاينكس (Linux)، أو ترميم (patch) لنظام التشغيل الحالي أو لبرنامج تطبيق حالي، أو ملف موسيقى MP3، أو ملف فيديو MPEG. في توزيع الملفات باستخدام بنية "زبون/خادم"، يجب أن يرسل الخادم نسخة من الملف إلى كل زبون من الزبائن المعنية، مما يُشكِّل عبئاً هائلاً على الخادم ويستهلك كمية كبيرة من الحيّز الترددي المخصص له. أما في توزيع الملفات ببنية النظائر، يمكن أن يعيد كل نظير توزيع أي جزء من الملف الذي استلمه إلى أيٍّ من النظائر الأخرى، وبذلك يساعد الخادم في عملية التوزيع. وفي وقت تأليف هذا الكتاب (ربيع عام 2007) كان البروتوكول الأكثر شعبية لتوزيع النظائر للملفات هو BitTorrent. وببعض التقديرات يمثل BitTorrent حوالي 30٪ من حركة مرور البيانات على شبكة العمود الفقري الرئيسة للإنترنت [CacheLogic 2007]. لقد طُوِّرهذا البروتوكول في الأصل من قِبَل برام كوهين (Bram Cohen)، وهناك الآن العديد من زبائن BitTorrent المستقلين والمتوافقين مع بروتوكول BitTorrent تماماً مثلما يوجد العديد من متصفّحات الويب المتوافقة مع بروتوكول HTTP. في هذا الجزء سنفحص أولاً القدرة الذاتية على التوسع (self-scalability) بسهولة لبنية النظائر في سياق توزيع الملفات، وبعدها سنصف بعض تفاصيل BitTorrent مبرزين خصائصه وميزاته الهامة.

قدرة بنية النظائر على التوسع

لمقارنة البنية المعمارية "زبون/خادم" ببنية النظائر المعمارية، وتوضيح القدرة الذاتية الطبيعية على التوسع في بنية النظائر، سنأخذ في الاعتبار الآن نموذجاً كمّياً بسيطاً لتوزيع ملف على مجموعة من النظائر عن طريق كلا النوعين للبنية المعمارية. يبين الشكل 2-24 اتصال الخادم والنظائر بالإنترنت. لِنَرمُز لمعدل التحميل لوصلة الوصول للخادم بـ u_s، ومعدل التحميل لوصلة الوصول للنظير i بـ u_i، ومعدل التنزيل لوصلة الوصول للنظير i بـ d_i. نرمز أيضاً لحجم الملف المراد توزيعه بـ F (بتات) ولعدد النظائر التي تريد الحصول على نسخة من الملف بـ N. نعرّف "زمن التوزيع" (distribution time) بأنه الزمن اللازم ليحصل كل نظير على

نسخة من الملف. في تحليلنا التالي لزمن التوزيع لكلٍّ من بنية "زبون/خادم" وبنية "النظائر" سنفترض لتبسيط التحليل أن "لُب" الإنترنت له حيّز ترددي وفير (وعموماً هذا الافتراض صحيح [Akella 2003])، مما يعني ضمناً أن عنق الزجاجة (bottleneck) يكمن في التوصل إلى الشبكة. نفترض أيضاً أن الخادم والزبائن لا يشاركون في أي تطبيقات شبكة أخرى، كي يكرّس كل الحيّز الترددي المتاح لهم بالكامل للتنزيل والتحميل اللازمين لتوزيع ذلك الملف.



الشكل 2-24 توضيح لمشكلة توزيع الملفات.

دعنا نحسب أولاً زمن التوزيع لبنية "زبون/خادم"، والذي نرمز له بـ D_{cs}. في بنية زبون/خادم لا تساعد النظائر في توزيع الملف. لنُبدِ الملاحظات التالية:

- يجب أن يرسل الخادم نسخة واحدة من الملف إلى كلٍّ من النظائر. وهكذا يجب أن يرسل الخادم NF بت. ولأن معدل إرسال الخادم u_s، فإن زمن توزيع الملف يجب أن يكون على الأقل $\frac{NF}{u_s}$.
- لنرمز لمعدل التنزيل للزبون صاحب أقل معدل بـ d_{min} أي أن

$$d_{min} = \min(d_1, d_2, ..., d_N)$$

لا يستطيع النظير صاحب معدل التنزيل الأدنى الحصول على كل بتات الملف (F بت) في أقل من $\frac{F}{d_{min}}$ ثانية. وهكذا يكون زمن التوزيع الأدنى على الأقل $\frac{F}{d_{min}}$.

وبدمج كلتا الملاحظتين معاً نحصل على

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}$$

وهو يمثل حداً أدنى للقيمة الصغرى لزمن التوزيع لبنية "زبون/خادم". في التمارين الموجودة في نهاية الفصل سوف يُطلَب منك إثبات أن الخادم يمكن أن يجدول إرساله لكي يتحقق الحد الأدنى في الواقع. لذا دعنا نأخذ هذا الحد الأدنى كزمن التوزيع الفعلي، أي أن:

$$D_{cs} = \max\left\{\frac{NF}{u_s}, \frac{F}{d_{min}}\right\}, \quad (2\text{-}1)$$

من هذه المعادلة يتبين أنه لقيم N الكبيرة بما فيه الكفاية، يكون زمن التوزيع لبنية "زبون/خادم" $\frac{NF}{u_s}$. وهكذا يزيد زمن التوزيع بشكلٍ خطي مع عدد النظائر N، فمثلاً إذا زاد عدد النظائر من أسبوع لآخر من ألف إلى مليون (أي ألف ضعف)، فإن زمن توزيع الملف إلى كل النظائر يزيد ألف ضعف كذلك.

دعنا نتناول الآن التحليل المناظر في حالة بنية النظائر، حيث يمكن أن يساعد كل نظير الخادم في توزيع الملف. بالتحديد عندما يتسلم نظير بعض بيانات الملف، يمكن أن يستخدم قدرته الخاصة على التحميل لإعادة توزيع البيانات إلى النظائر الأخرى. يُعتبر حساب زمن التوزيع لبنية النظائر أكثر تعقيداً بعض الشيء مقارنة ببنية "زبون/خادم"، لأن زمن التوزيع يعتمد على الكيفية التي يوزّع بها كل نظير أجزاء الملف إلى النظائر الأخرى. على الرغم من ذلك يمكن الحصول على تعبير بسيط للقيمة الصغرى لزمن التوزيع [Kumar 2006]. وسعياً لهذا الهدف دعنا نُبدي الملاحظات التالية أولاً:

- في بداية التوزيع يكون الخادم فقط هو الذي يمتلك الملف. ولتوزيع ذلك الملف إلى مجموعة النظائر، يجب أن يرسل الخادم كل بت من الملف مرة واحدة على الأقل إلى وصلة الوصول للشبكة لديه. وهكذا تكون القيمة الصغرى لزمن التوزيع على الأقل $\frac{F}{u_s}$. (على خلاف بنية "زبون/خادم"، قد لا يحتاج الخادم لإرسال كل بت أكثر من مرة، لأن النظائر قد تعيد توزيع البت فيما بينها).
- كما هو الحال مع بنية "زبون/خادم"، لا يستطيع النظير الذي له أقل معدل تنزيل الحصول على كل بتات الملف (F بت) في أقل من $\frac{F}{d_{\min}}$ ثانية. وهكذا تكون القيمة الصغرى لزمن التوزيع تساوي $\frac{F}{d_{\min}}$ على الأقل.
- أخيراً لاحظ أن قدرة الإرسال الكلية للنظام ككل تساوي معدل التحميل من الخادم بالإضافة إلى معدل إرسال كل فرد من النظائر، أي

$$u_{total} = u_s + u_1 + \ldots + u_N$$

يجب أن يُحمِّل النظام F بت إلى كلٍّ من النظائر التي عددها N، ومن ثم يُحمِّل النظام ما مجموعه NF بت. وهذا لا يمكن تحقيقه بمعدل أسرع من u_{total}. وهكذا تكون القيمة الصغرى لزمن التوزيع أيضاً على الأقل

$$\frac{NF}{u_s + u_1 + \ldots + u_N}$$

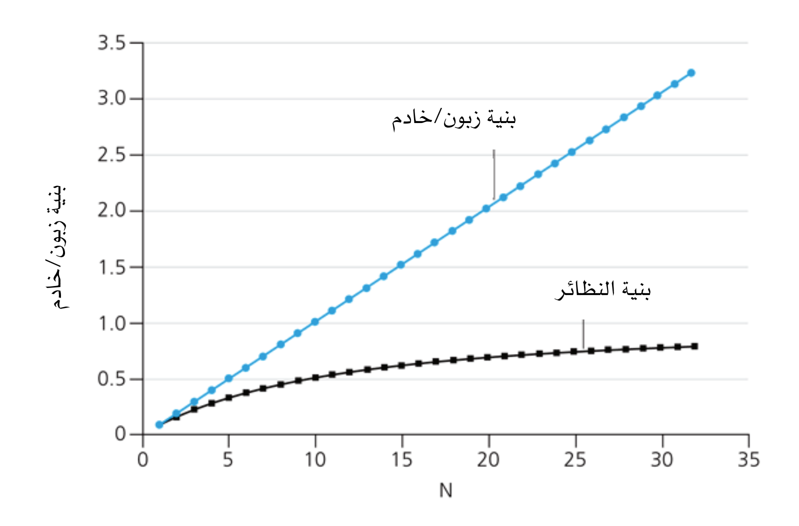
بدمج هذه الملاحظات الثلاث معاً نحصل على القيمة الصغرى لزمن التوزيع لبنية النظائر، (نشير إليه بـ D_{P2P}).

$$D_{P2P} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}, \quad (2\text{-}2)$$

تعطي هذه المعادلة حداً أدنى للقيمة الصغرى لزمن التوزيع لبنية النظائر. وبالتالي إذا تخيلنا أن كل نظير يمكن أن يعيد توزيع البت بمجرد تسلّمها، فعندئذ توجد طريقة لإعادة التوزيع تحقق هذا الحد الأدنى في واقع الأمر [Kumar 2006]. (سوف نُثبت حالة خاصة لهذه النتيجة في تمارين نهاية الفصل). في الواقع عندما يعاد توزيع قطع من الملف بدلاً من البتات كلٍّ على حدة، تعتبر المعادلة 2-2 تقريباً جيداً للقيمة الفعلية الصغرى لزمن التوزيع. ولذا سنعتبر أن الحد الأدنى المعطى بالمعادلة 2-2 يمثل القيمة الفعلية الصغرى لزمن التوزيع، أي

$$D_{P2P} = \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^{N} u_i}\right\}, \quad (2\text{-}3)$$

يُقارن الشكل 2-25 القيمة الصغرى لزمن التوزيع لكلٍّ من بنية "زبون/خادم" وبنية النظائرعلى افتراض أن كل النظائر لها نفس معدل التحميل u.
في الشكل 2-25 افترضنا أن $\frac{F}{u}$ = 1 ساعة، $u_s = 10u$، $d_{\min} \geq u_s$. أي أنه بوسع النظيرإرسال الملف بكامله في ساعة واحدة، وأن معدل إرسال الخادم يساوي عشر مرات معدل إرسال النظير، وللتبسيط افترضنا أن معدل التنزيل للنظائر كبير بما فيه الكفاية بحيث لا يكون له تأثير. نرى من الشكل 2-25 أن زمن التوزيع يزيد بشكلٍ خطي لبنية "زبون/خادم"، وبدون حد مع زيادة عدد النظائر. ولكن زمن التوزيع الأدنى لبنية النظائر ليس فقط دائماً أقل من زمن التوزيع لبنية "زبون/خادم"، وإنما أيضاً أقل من ساعة واحدة لأي عدد من النظائر N. ومن ثم يمكن أن تكون تطبيقات النظائر ذاتية التوسع بسهولة. إن هذه القدرة على التعامل الجيد مع التوسع هي نتيجة مباشرة لكون النظائر بمثابة نقاط لإعادة توزيع (redistributors) البيانات بالإضافة إلى كونها مستهلكة (consumers) لتلك البيانات في نفس الوقت.



الشكل 2-25 زمن التوزيع لبنية زبون/خادم وبنية النظائر.

بروتوكول BitTorrent

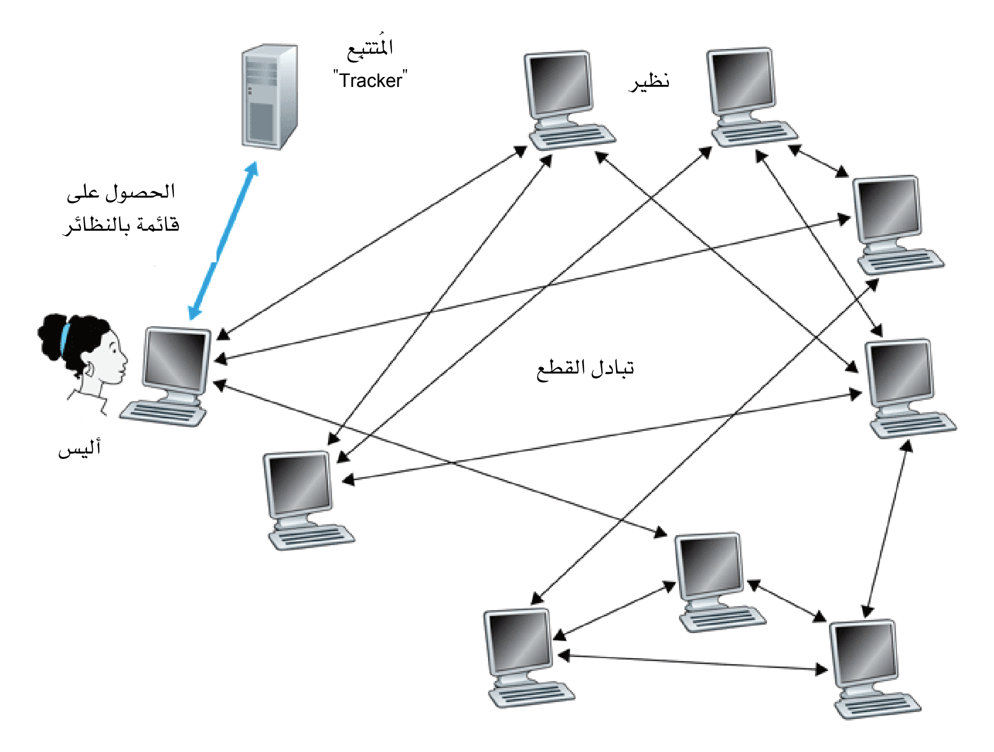
يُعتبر BitTorrent بروتوكول نظائر شائع الاستعمال لتوزيع الملفات [BitTorrent 2007]. في مصطلحات BitTorrent تسمى مجموعة النظائر التي تشارك في توزيع ملف معين "سيلاً" (torrent). تُنزِّل النظائر الموجودة في "السيل" قطعاً متساوية من الملف من أحدها للآخر، في العادة يكون حجم القطعة 256 كيلوبايت. وعندما ينضم نظير إلى "السيل" في البداية، لا يكون لديه قطع من الملف، ولكن بمرورالوقت يقوم بتجميع المزيد والمزيد من القطع. وبينما يُنزّل قطعاً فإنه يُحمِّل أيضاً قطعاً إلى النظائر الأخرى. عندما يحصل نظير على كامل الملف، قَد يغادر السيل (بشكلٍ أناني)، أو يبقى في السيل (بدافع مساعدة الغير) ويُواصل إرسال القطع إلى النظائر الأخرى. أيضاً قد يترك أي نظير السيل في أي وقت بعد تنزيل مجموعة جزئية من القطع فقط، ثم ينضم ثانية إلى السيل لاحقاً.

دعنا الآن نلقي نظرة أكثر تفحصاً على كيفية عمل BitTorrent. نظراً لأن BitTorrent بروتوكول معقّد نوعاً ما، فسوف نصف آلياته الهامة فقط، ونغض

الطرف عن بعض التفاصيل الأخرى لنتمكن من رؤية الغابة من خلال الأشجار. يتضمن كل سيل عقدة بنية تحتية يطلق عليها "المقتفي" (tracker). عندما ينضم نظير إلى السيل، يُسجّل نفسه بالمقتفي ويُخبر المقتفي بشكلٍ دوري بأنه ما زال في السيل. بهذه الطريقة يتابع المقتفي النظائر المشاركة في السيل. وربما يضم سيلٌ ما المئات أو الآلاف من النظائر التي تشارك فيه في وقت من الأوقات.

كما هو موضح في الشكل 2-26، عندما ينضم نظير جديد مثل أليس إلى السيل، يختار المقتفي بشكلٍ عشوائي مجموعة جزئية من النظائر (لنقُل 50 نظيراً مثلاً) من مجموعة النظائر المشاركة، ويُرسل عناوين IP الخاصة بتلك النظائر الـ 50 إلى أليس. مع امتلاك هذه القائمة من النظائر، تحاول أليس إنشاء توصيلات TCP في نفس الوقت مع كل النظائر على تلك القائمة. دعنا ندعو كل النظائر التي نجحت أليس في إنشاء توصيلة معها "النظائر المجاورة". (في الشكل 2-26 لدى أليس ثلاثة نظائر مجاورة فقط، وعادةً يكون لديها أكثر من ذلك). ومع مرور الوقت قد تغادر بعض تلك النظائر بينما قد تحاول نظائر أخرى (خارج الخمسين الأولى) إنشاء توصيلات TCP مع أليس، لذا ستتغير مجموعة النظائر المجاورة بمرور الوقت.

في أي وقت سيكون لدى كل نظير مجموعة جزئية من قطع الملف، ويكون عند النظائر المختلفة مجموعات جزئية مختلفة. وبشكلٍ دوري ستسأل أليس كلاً من نظائرها المجاورة (على توصيلات TCP) عن قائمة القطع التي وصلتهم. وإذا كانت أليس لديها عدة جيران عددهم L، فستحصل على عدد L من قوائم القطع. وبناءً على تلك المعرفة، ستصدر أليس الطلبات (مرة ثانية على توصيلات TCP) للحصول على القطع التي ليست لديها حالياً.



الشكل 2-26 توزيع الملفات باستخدام بروتوكول BitTorrent.

لذلك فإنه في أي لحظة سيكون لدى أليس مجموعة جزئية من القطع وستعرف أي قطع تتوافر عند جيرانها. وبهذه المعلومات سيكون عليها اتخاذ قرارين مهمين هما: ما القطع التي يجب أن تطلبها أولاً من جيرانها؟، وإلى أي جيرانها يجب أن تُرسِل القطع المطلوبة؟ لتحديد القطع التي تطلبها أليس تستخدم أسلوباً يطلق عليه "الأندر أولاً" (rarest first). تتلخص الفكرة في أن تحدد من بين القطع التي ليست لديها القطع "الأندر" بين جيرانها (القطع ذات النسخ الأقل تكرراً بين جيرانها)، وبعد ذلك تطلب تلك القطع أولاً. بهذا الأسلوب يعاد توزيع القطع الأندر بسرعة أكبر، مما يؤدي إلى تساوي أعداد نسخ كل قطعة في السيل تقريباً. ولتحديد أي الطلبات ترد عليها أليس، يستخدم BitTorrent خوارزمية تجارية ذكية تتلخص فكرتها الأساسية في أن تعطي أليس أولوية للجيران الذين

يزودونها بالبيانات حالياً بأعلى معدل. وبشكلٍ محدد تقيس أليس بشكلٍ مستمر المعدل الذي تستلم به البيانات من كلٍّ من جيرانها، ثم تختار النظائر الأربعة التي تغذيها بالبيانات بأعلى معدل، وتتبادل معها القطع. وبعد كل عشر ثوانٍ، تعيد أليس حساب المعدلات ومن المحتمل إجراء تعديل على مجموعة النظائر الأربعة الحالية. من المهم ملاحظة أن كل 30 ثانية تلتقط أليس جاراً إضافياً واحداً بطريقة عشوائية، وترسل له قطعاً من الملف. دعنا نطلق على ذلك النظير المختارعشوائياً بوب. نظراً لأن أليس ترسل البيانات إلى بوب، فقد تصبح واحداً من النظائر الأربعة المُحمِّلة (uploaders) على القمة لدى بوب. وفي هذه الحالة يبدأ بوب بإرسال البيانات إلى أليس. إذا كان المعدل الذي يرسل به بوب البيانات إلى أليس عالياً بما فيه الكفاية، يمكن أن يصبح بوب بدوره واحداً من النظائر الأربعة المُحمِّلة على القمة لدى أليس. أي أن كل 30 ثانية ستختار أليس بشكلٍ عشوائي شريكاً جديداً لتبادل البيانات معه، ثم تستهل التعامل معه. وإذا أصبح النظيران راضيين عن هذا التعامل، فسوف يضع كلٌّ منهم الآخر في قوائم "الأربعة العليا" لديه ويستمر بالتعامل معه إلى أن يجد أحدهما شريكاً أفضل. ونتيجة لذلك تتجه النظائر المتوافقة في معدلات الإرسال إلى العثور على بعضها البعض. في نفس الوقت يسمح اختيار الجار العشوائي أيضاً لنظائر جديدة بالحصول على قطع، ليكون لديهم شيء يتبادلونه. كل النظائر المتجاورة الأخرى غير النظائر الخمسة (أربعة نظائر "عليا" ونظير المجس (probing peer) تصبح "مخنوقة"، بمعنى أنها لا تتلقى أي قطع من أليس .

توجد مشكلة شائعة في مشاركة النظائر للملفات، ألا وهي مشكلة "الركوب المجاني" (free-riding)، وفيها يُنزِّل نظير ملفات من نظام مشاركة الملفات دون أن يرسل هو ملفات. من المفترض أن بروتوكول BitTorrent يتغلب على المشكلة باستخدام خوارزمية "التبادل التجاري"، لأنه لكي تتمكن أليس من تنزيل القطع من بوب بمعدل مقبول لفترة زمنية طويلة، يجب في نفس الوقت أن ترسل القطع إلى بوب بمعدل مقبول. يمتلك BitTorrent عدة آليات مثيرة أخرى لن نناقشها هنا مثل تقطيع الملف إلى أجزاء، والمعالجة بطريقة خط الأنابيب

(pipelining)، والاختيار العشوائي الأول (random first selection)، ونمط نهاية اللعبة (end game mode)، ومانع الوقف (anti-snubbing) [Cohen 2003].

2-6-2 البحث عن معلومات في مجتمع النظائر

يُشكِّل "فهرس المعلومات" أحد المكونات الهامة في العديد من تطبيقات النظائر – بمعنى ربط المعلومات مع مواقع المضيفات. في مثل تلك التطبيقات تقوم النظائر بتعديل الفهرس والبحث فيه بطريقة ديناميكية. نظراً لأن فكرة "ربط المعلومات مع مواقع المضيفات" قد تبدو مجرّدة نوعاً ما، لنُلقِ نظرة على مثالين محددين.

- يوجد في نظام مشاركة النظائر للملفات عادةً عدد كبير من النظائر المشاركة، وكل نظير لديه ملفات يشارك بها كملفات MP3 والفيديو والصور والبرامج. يتضمن نظام مشاركة النظائر للملفات فهرساً يتعقب بطريقة ديناميكية الملفات التي توفرها النظائر للمشاركة. ولكل نسخة من كل ملف متاح للمشاركة بين مجموعة النظائر، يحتفظ الفهرس بسجل يربط بين المعلومات عن النسخة (مثلاً إذا كان ملف MP3 لأغنية، تكون المعلومات عنوان الأغنية، واسم الفنان، وهكذا) وعنوان IP للنظير الذي لديه تلك النسخة. يُعدّل الفهرس بطريقة ديناميكية بينما تأتي النظائر وتروح، وتحصل النظائر على نسخ جديدة من الملفات. على سبيل المثال عندما ينضم نظير إلى النظام يُخبِر الفهرس بالملفات الموجودة لديه. وعندما يريد مُستخدِم معين مثل أليس الحصول على ملف معين تبحث في الفهرس لتحديد النظائر التي لديها نسخ من الملف المطلوب. بعد الحصول على مواقع تلك النظائر سيكون بوسع أليس تنزيل الملف منها. وعندما يصبح لديها الملف بكامله، يتم تعديل الفهرس ليشمل نسخة أليس الجديدة من الملف.
- في تطبيق الرسائل الفورية (instant messaging) يوجد فهرس يربط ما بين أسماء المُستخدِمين ومواقعهم (أي عناوين IP لهم). ولفهم أهمية الفهرس في

هذا التطبيق، افترض أن كلاً من المُستخدِمين BeautifulAlice وHandsomeBob موجود على قائمة "الرفقاء" للآخر. عندما يبدأ HandsomeBob برنامج زبون الرسائل الفورية على مضيف بعنوان IP = X، سوف يخبر زبونه الفهرس بأن HandsomeBob على الإنترنت وعنوانه X. لاحقاً عندما تبدأ BeautifulAlice برنامج المراسلة الفورية لديها، حيث إن HandsomeBob على قائمة رفقائها، يبحث الزبون لديها في الفهرس عن HandsomeBob ويكتشف بأن HandsomeBob على الإنترنت على العنوان X. يمكن أن تنشئ BeautifulAlice عندئذ توصيلة TCP إلى المضيف في العنوان X وتبدأ بالمراسلة الفورية مع HandsomeBob. وبالإضافة إلى المراسلة الفورية تستخدم الكثير من التطبيقات الأخرى اليوم فهرساً لتتبع التواجد، بما في ذلك أنظمة هواتف الإنترنت (انظر الجزء 2-6-3).

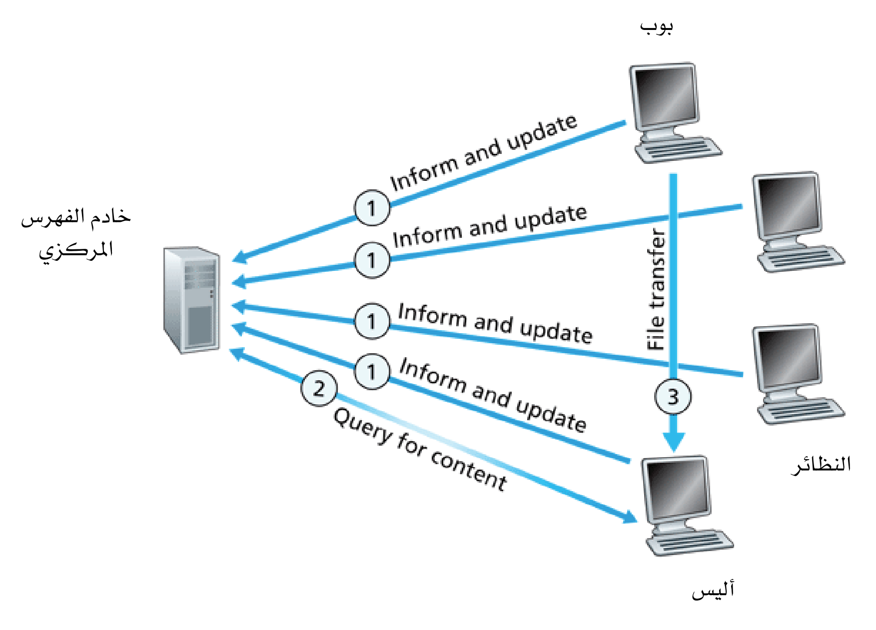
ونذكر باختصار هنا أن نظام BitTorrent يمثل بروتوكولاً لتوزيع الملفات فقط، ولا يوفر أي وظائف لفهرسة الملفات والبحث عنها.

سنناقش فيما يلي ثلاث طرق لتنظيم الفهرس والبحث فيه في مجتمع النظائر. ولنكون أكثر تحديداً، سوف نقوم بذلك في سياق البحث عن ملف في نظام مشاركة النظائر للملفات. ولكننا نؤكد على أن هذه المناقشة تنطبق على حدٍّ سواء على البحث عن أي نوع من المعلومات في مجتمع النظائر.

الفهرس المركزي

يعتبر استخدام فهرس مركزي أحد الطرق البسيطة لتحديد مكان ملف (كما في Napster والذي يمثل الاستخدام التجاري الأول لمشاركة النظائر للملفات على نطاق واسع). في هذا التصميم يتم توفير خدمة الفهرس من قِبَل خادم كبير (أو مزرعة خادمات). كما هو موضح في الشكل 2-27، عندما يبدأ مُستخدِم تطبيق مشاركة النظائر للملفات، فإنه يخبر خادم الفهرس بعنوان IP له وأسماء الملفات المتوفرة لديه للمشاركة (على سبيل المثال عناوين كل ملفات MP3 المُخزَّنة عليه). يجمع خادم الفهرس تلك المعلومات من كل نظير فعّال ومن ثم يُنشئ فهرساً

مركزياً ديناميكياً يربط ما بين كل نسخة ملف ومجموعة من عناوين IP. لاحظ أن نظام النظائر لمشاركة الملفات الذي يستخدم فهرساً مركزياً هو في الحقيقة نظام "هجين" من بنية النظائر وبنية زبون/خادم. يستخدم توزيع الملف بنية النظائر ولكن يستخدم البحث عنه بنية "زبون/خادم". يمكن أن توجد مثل هذه البنية المعمارية الهجينة في عدد من التطبيقات اليوم، بما في ذلك العديد من برامج المراسلة الفورية.



الشكل 2-27 الفهرس المركزي.

يُعدُّ استخدام "فهرس مركزي" لتحديد مكان المعلومات طريقة مباشرة ذات مفهوم واضح، غير أن لها عدة عيوب:

- نقطة فشل وحيدة: إذا تعطل خادم الفهرس فإن التطبيق بكامله يتعطل. حتى إذا استخدمنا "مزرعة خادمات"، يمكن أن تتعطل توصيلات الإنترنت إلى مزرعة الخادمات، مما يسبب تعطل التطبيق بأكمله.

- أداء عنق الزجاجة وبنية تحتية مكلفة: في نظام نظائر كبير يضم مئات الآلاف من المُستخدِمين المُوصَّلين، يجب أن يحتفظ الخادم المركزي بفهرس ضخم ويجب أن يرد على آلاف الاستفسارات كل ثانية. في الحقيقة في عام 2000 عندما كان تطبيق Napster الأكثر شعبية عانى من مشاكل ازدحام حركة المرورعند خادمه المركزي.
- انتهاك حقوق النشر: رغم أن هذا الموضوع يقع خارج مجال هذا الكتاب، فإننا نذكر باختصار هنا أن شركات تسجيلات الصوت والفيديو كانت مشغولة وقلقة بشأن استخدام نظام مشاركة النظائر للملفات لأنه يسمح للمُستخدِمين بالحصول بسهولة على "المحتوى المحفوظ الحقوق" مجاناً. ولمناقشة ممتازة حول قوانين حقوق النشر المرتبطة بنظام النظائر، طالع [von Lohmann 2003]. عندما تملك شركة لمشاركة النظائر للملفات خادم فهرس مركزي، قد يؤدي اتخاذ إجراء قانوني ما إلى إغلاق خادم الفهرس. في حين يصعب ذلك في البنية اللامركزية.

فيضان الاستفسار

على الطرف المعاكس للفهارس المركزية توجد الطريقة اللامركزية تماماً وهي "فيضان الاستفسار". وقد تجسدت هذه الطريقة في بروتوكول Gnutella الأصلي، وفيها يكون الفهرس موزعاً بالكامل على مجتمع النظائر حيث يحتوي كل نظير فقط على فهرس للملفات التي يوفرها للمشاركة وليس أي ملفات أخرى.

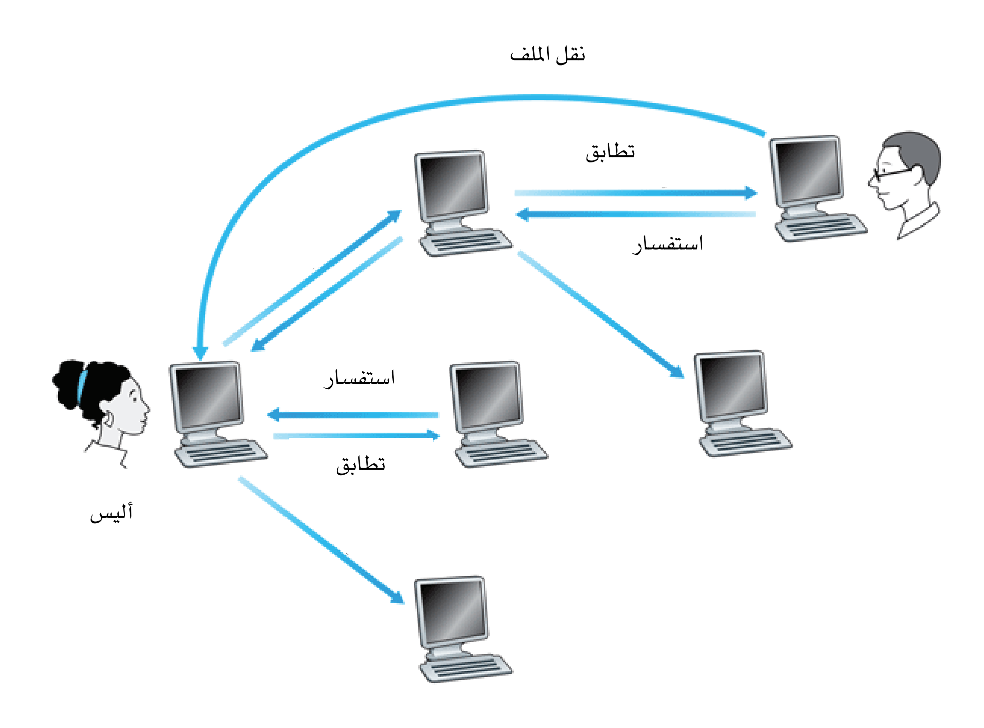
وتُشكِّل النظائر شبكة منطقية مجرّدة يطلق عليها "شبكة إضافية" (overlay network)، ويمكن تعريفها باستخدام مصطلحات "نظرية الأشكال البيانية" (graph theory) كالتالي: إذا كان النظير X يحتفظ بتوصيلة TCP مع نظير آخر Y، نقول بأن هناك "رابط" (edge) بين X وY. ويُعرَف الشكل البياني الذي يضم كل النظائر النشطة وروابط التوصيل (توصيلات TCP الحالية) بالشبكة الإضافية. لاحظ أن "الرابط" ليس وصلة اتصال مادية، وإنما هو وصلة

مجرّدة قد تشمل تحتها عدة وصلات مادية. فعلى سبيل المثال قد يمثل "رابط" في الشبكة الإضافية توصيلة TCP بين نظير في ليثوانيا وآخر في البرازيل.

ورغم أن مثل تلك الشبكة الإضافية قد تتكون من مئات الآلاف من النظائر المشاركة، فإنه في العادة يتصل النظير بعدد قليل من النظائر الأخرى (عادة أقل من عشرة) ضمن الشبكة الإضافية، كما هو موضح في الشكل 2-28. سوف نوضّح لاحقاً كيف يمكن أن تُبنَى الشبكة الإضافية وتعدل بينما تنضم إليها نظائر وتغادر أخرى. دعنا الآن نفترض أن الشبكة الإضافية موجودة ولنركز على كيفية تحديد نظير لمكان محتوى وكيفية حصوله على ذلك المحتوى.

في هذا التصميم ترسل النظائر الرسائل إلى النظائر المجاورة في الشبكة الإضافية على توصيلات TCP الموجودة مسبقاً. مثلاً عندما تريد أليس تحديد مكان ملف "Network Love"، سوف يرسل برنامج الزبون لديها رسالة استفسار تتضمن الكلمات الدليلية "Network Love" إلى كل جيرانها، ومن ثم يسلّم كل واحد منهم الاستفسار إلى كل جيرانه، وهكذا. يوضح الشكل 2-28 عملية "فيضان الاستفسار" هذه. عندما يتلقى نظير استفساراً، يفحص ليرى ما إذا كانت الكلمة الدليلية تطابق أياً من الملفات المتوفرة لديه للمشاركة. وإذا حدث تطابق يرسل النظير رسالة "query-hit" إلى أليس تتضمن اسم وحجم ذلك الملف. تتبع تلك الرسالة الطريق العكسي لرسالة الاستفسار، مستخدمة بذلك توصيلات TCP الموجودة مسبقاً. وبهذا الأسلوب تكتشف أليس النظائر التي تمتلك نسخة من الملف الذي تريده.

رغم أن هذا التصميم اللامركزي بسيط ورائع، إلا أنه عادةً ما يُنتقَد لعدم قابليته للتوسع. وبالتحديد حينما يبدأ نظير استفساراً، يتدفق الاستفسار لكل نظير آخر في الشبكة الإضافية بكاملها، مما يولد كمية هائلة من حركة البيانات بين النظائر في الشبكة التحتية (كالإنترنت) التي تصل ما بين النظائر. طوّر مصممو Gnutella حلاً لهذه المشكلة باستعمال "فيضان الاستفسار محدود المجال". وفيه عندما تبعث أليس رسالة الاستفسار الأولى تُضبَط قيمة حقل عدد



الشكل 2-28 فيضان الاستفسار.

النظائر (peer count) في الرسالة عند حد معين (مثلاً 7). كل مرة تصل رسالة الاستفسار إلى نظير جديد ينقص النظير قيمة هذا الحقل بمقدار واحد قبل إعادة إرساله إلى جيرانه في الشبكة الإضافية. وعندما يستلم نظير استفساراً وتكون قيمة هذا الحقل قد أصبحت "صفراً"، فإنه يتوقف عن إرسال هذا الاستفسار. بهذا الأسلوب يتم حصر الفيضان في منطقة الشبكة الإضافية المحيطة بالنظير الذي بدأ الاستفسار محدود المجال. واضح أن هذا من شأنه الحد من حركة مرور الاستفسارات، لكنه يقلل أيضاً من مجال البحث، وعليه فمن المحتمل ألا يتمكن نظير يريد محتوى معين من تحديد مكانه، بالرغم من وجود ذلك المحتوى في مكان ما في مجتمع النظائر.

تعتبر معالجة النظائر التي تنضم إلى الشبكة أو تغادرها قضية أساسية في الشبكات الإضافية. باستخدام تصميم Gnutella الأصلي كمثال فإن الإجراءات التي تُتخَذ لتعديل الشبكة عندما تنضم نظائر جديدة إليها كالنظير X:

1. يجب أولاً أن يجد النظير X نظيراً آخر موجوداً حالياً في الشبكة الإضافية. لحل معضلة البدء (bootstrap) هذه يمكن أن يحتفظ X بقائمة النظائر (عناوين IP) التي "تعمل" في أغلب الأحيان في الشبكة الإضافية. كبديل لذلك يمكن أن يتصل X بموقع "المقتفي" (tracker) والذي يحتفظ بمثل تلك القائمة (كما في BitTorrent).
2. بمجرد أن يحصل X على مثل تلك القائمة، يحاول بدء توصيلة TCP مع النظائر التي على القائمة الواحد تلو الآخر حتى يتحقق اتصال مع واحد منها، وليكن Y.
3. بعد إنشاء توصيلة TCP بين X وY، يمكن أن يرسل النظير X رسالة "ping" إلى Y تتضمن حقل عدد النظائر. وفور استلام Y لتلك الرسالة، يرسلها إلى كل جيرانه في الشبكة الإضافية. وتواصل النظائر إرسال الرسالة حتى يصبح حقل عدد النظائر صفراً.
4. وحينما يستلم نظير Z رسالة البينج، يرد بإرسال رسالة "pong" (تتضمن عنوان IP لـ Z) خلال الشبكة الإضافية إلى X.
5. بعد استلام X رسائل pong، يعرف عناوين IP للعديد من النظائر في الشبكة الإضافية. عندئذ يمكنه أن يبدأ توصيلات TCP ببعض تلك النظائر الأخرى، وبذلك يُنشئ "روابط" متعددة بينه وبين الشبكة الإضافية.

سوف نستكشف الإجراءات التي يمكن أن تتخذها الشبكة الإضافية فور مغادرة نظير في تمارين نهاية الفصل.

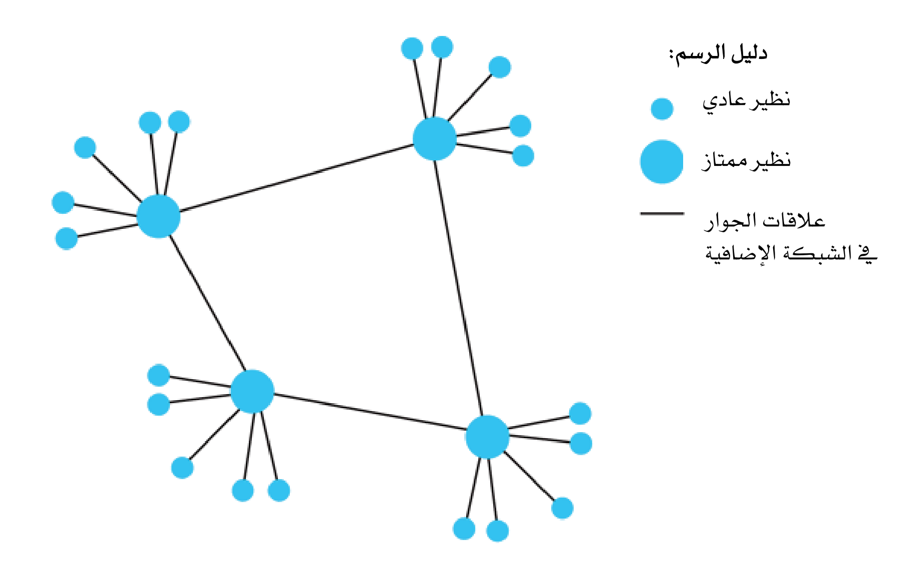
حتى الآن غطينا الخصائص الضرورية لفيضان الاستفسار وبناء الشبكة الإضافية ديناميكياً. وباختصار فإن "فيضان الاستفسار" أسلوب بسيط وموزّع بين النظائر يسمح لمُستخدِم بالسؤال عن المعلومات التي توجد في النظائر القريبة (أي

ضمن عدد صغير من القفزات (hops) في الشبكة الإضافية). طبّق تصميم بروتوكول Gnutella الأصلي "فيضان الاستفسار" كما وصفنا أعلاه. وعلى مرِّ السنين تطوّر البروتوكول بشكلٍ ملحوظ، والآن يستغل عدم تجانس النظائر في نظام النظائر لمشاركة الملفات. ويبقى بروتوكول Gnutella شائعاً جداً اليوم، كما أنه يُستخدَم في زبون نظام النظائرالشائع LimeWire.

البناء الهرمي للشبكات الإضافية

عرفنا أن "الفهرس المركزي" و"فيضان الاستفسار" طريقتان متعارضتان تماماً لتحديد مكان المعلومات. هناك طريقة ثالثة سوف نشير إليها "بالتصميم الهرمي للشبكات الإضافية"، وهي تجمع أفضل ميزات الطريقتين السابقتين. ابتُكر "التصميم الهرمي للشبكات الإضافية" أولاً من قِبَل FastTrack (وهو نظام نظائر لمشاركة الملفات استخدم في عدد من التطبيقات على مرِّ السنين، مثل Kazaa وMorpheus). كما يستخدمه أيضاً نظام Gnutella الحديث، رغم أنه يختلف عن النظام الموصوف هنا.

كما هو الحال مع "فيضان الاستفسار" لا يكرس "التصميم الهرمي للشبكات الإضافية" خادماً (أو مزرعة خادمات) لتتبع وفهرسة الملفات. ومع ذلك فبخلاف فيضان الاستفسار ليست كل النظائر متماثلة في التصميم الهرمي للشبكات الإضافية. بالتحديد يطلق على النظائر المتاحة غالباً والتي تتصل بالإنترنت بوصلات لها سعة إرسال عالية نظائر ممتازة (عليا) (super peers) ويكون لها مسؤوليات أكبر. وكما هو موضح في الشكل 2-29، إذا لم يكن النظير "نظيراً ممتازاً"، فإنه يكون "نظيراً عادياً" وينسب ابناً لنظير ممتاز. قد يكون للنظير الممتاز بضعة مئات من النظائر العادية كأبناء.



الشكل 2-29 البناء الهرمي للشبكات الإضافية.

يُنشئ النظير الجديد أولاً توصيلة TCP بأحد النظائر الممتازة، ثم يُخطره بكل الملفات التي يوفرها للمشاركة. يسمح ذلك للنظير الممتاز بالاحتفاظ بفهرس يتضمن هوية كل ملفات النظائر الأبناء له، كما يتضمن بيانات ما وراء البيانات (meta-data) (أي بيانات عن بيانات أخرى) حول تلك الملفات، وعناوين IP المناظرة للأبناء الذين لديهم تلك الملفات. بهذه الطريقة يصبح كل نظير ممتاز فهرساً "مصغراً" (mini-index).

بالمقارنة مع الفهرس المركزي الذي ناقشناه في بداية هذا الجزء، لا يعتبر النظير الممتاز خادماً مكرّساً، وإنما هو نظير عادي، وعادة ما يوجد في حي سكني أو حرم جامعي. وإذا ما عزل كل نظير ممتاز وأبناؤه عن الشبكة فإن ذلك سيحد من كمية المحتوى المتوفرة لأي نظير بشكلٍ ملحوظ. لعلاج هذا التقييد ترتبط النظائر الممتازة فيما بينها بتوصيلات TCP لتكوين شبكة إضافية بينها. وبهذا يمكن أن ترسل النظائر الممتازة الاستفسارات إلى النظائر

الممتازة المجاورة. هذه النظرة مشابهة لفيضان الاستفسار، لكنه فيضان محدود المجال يتم داخل الشبكة الإضافية بين النظائر الممتازة.

عندما يريد نظير إجراء بحث باستخدام كلمات دليلية (keywords)، يرسل استفساراً يتضمن تلك الكلمات إلى نظيره الممتاز. فيرد النظير الممتاز بعناوين IP لنظائره الأبناء الذين لديهم ملفات تتوافق موصّفاتها (descriptors) مع تلك الكلمات الدليلية (مع مُعرِّفات (identifiers) لتلك الملفات). قد يرسل النظير الممتاز الاستفسار أيضاً إلى واحد أو أكثر من النظائر الممتازة الأخرى المجاورة. إذا تلقى نظير مجاور مثل هذا الاستفسار، فإنه يرد أيضاً بعناوين IP لنظائره الأبناء الذين لديهم ملفات مطابقة. تتبع الردود من النظائر الممتازة الطريق العكسي في الشبكة الإضافية.

يستغل التصميم الهرمي للشبكات الإضافية عدم التجانس الطبيعي للنظائر بتعيين عدد قليل من النظائر ذات القدرات الأكبر كنظائر ممتازة، لتكوين الطبقة العليا (top tier) لهرم الشبكة الإضافية، كما هو موضح في الشكل 2-29. بالمقارنة مع فيضان الاستفسار محدود المجال (كما في تصميم Gnutella الأصلي)، يسمح التصميم الهرمي للشبكات الإضافية بفحص "التطابق" مع عدد كبير جداً من النظائر، بدون توليد كمية مفرطة من حركة مرور الاستفسار [Liang 2005].

قبل أن ننهي مناقشتنا للبحث عن المعلومات في تطبيقات النظائر، نذكر باختصار تصميماً مهماً آخر يطلق عليه الجدول الهاش المُوزَّع (Distributed (DHT (Hash Table [Stoica 2001; Rowstron 2001; Ratnasamy 2001; Zhao 2004; Maymounkov 2002; Garces-Erce 2003]. إن المناقشة المستفيضة لهذا الموضوع تقع خارج مجال هذا الكتاب. لكننا نذكر هنا أن (1) يُنشئ DHT فهرساً غير مركزي تماماً يربط ما بين مُعرِّفات الملفات ومواقعها، (2) يسمح للمُستخدِم بتحديد كل مواقع الملف (من حيث المبدأ) بدون توليد كمية مفرطة من حركة مرورالبحث. وقد لاقى DHT اهتماماً كبيراً في المحيط البحثي، وهو مستخدم في

Overnet والذي يمثل جزءاً محورياً من تطبيق مشاركة الملفات الشائع eMule [Liang 2006].

2-6-3 دراسة حالة: الاتصال الهاتفي للنظائر عبر الإنترنت باستخدام سكايب (Skype)

يُعتبر سكايب تطبيق نظائر شائع الاستخدام بشكلٍ كبير، حيث يستخدمه في أغلب الأحيان من سبعة إلى ثمانية ملايين مُستخدِم متصلون به في أي لحظة. وبالإضافة إلى أنه يوفر خدمة اتصال هاتفي عبر الإنترنت بين أجهزة الحاسب (PC-to-PC)، فهو يقدم خدمات الاتصال الهاتفي من الحاسب للهاتف (PC-to-Phone)، والاتصال الهاتفي من الهاتف للحاسب (Phone-to-PC)، ومؤتمرات الفيديو بين أجهزة الحاسب (PC-to-PC). قام بتأسيس سكايب نفس الأشخاص الذين بنوا FastTrack وKazaa وقد اشترته eBay عام 2005 مقابل 2.6 بليون دولار.

يستخدم سكايب تقنيات النظائر في عدد من الطرق الإبداعية، ويُصوّر بشكلٍ رائع كيف يمكن استخدام النظائر في التطبيقات التي تتجاوز "توزيع المحتوى" و"مشاركة الملفات". كما هو الحال مع المراسلة الفورية، فإن الاتصال الهاتفي عبر الإنترنت PC-to-PC هو في الأصل نظام نظائر، ذلك لأنه في قلب التطبيق يتصل أزواج من المُستخدِمين (وبمعنى آخر النظائر) مع بعضهم في الوقت الحقيقي. لكن سكايب يستخدم أيضاً تقنيات النظائر لوظيفتين مهمتين أخريين: تحديد موقع المُستخدِم واجتياز الـ NAT.

ليست بروتوكولات سكايب فقط ذات ملكية خاصة، ولكن أيضاً كل رزم سكايب المُرسَلة (رزم الصوت والتحكم) مشفّرة تشفيراً سرياً. وعلى الرغم من ذلك فمن خلال موقع الويب الخاص به وعدد من دراسات القياسات تعلم الباحثون كيف يعمل سكايب عموماً [Baset 2006; Guha 2006; Chen 2006; Suh 2006; Ren 2006]. وكما هو الحال مع FastTrack، تنتظم العُقَد في Skype في ترتيب هرمي للشبكة الإضافية، وكل نظير مصنّف كنظير ممتاز أو نظير عادي. يتضمن Skype فهرساً يربط أسماء مُستخدِمي سكايب مع عناوين IP

الحالية (وأرقام المنافذ)، وهذا الفهرس موزّع على النظائر الممتازة. عندما تريد أليس محادثة بوب، فسوف يبحث زبون Skype لديها في الفهرس الموزّع لتحديد عنوان IP الحالي لبوب. ولأن نظام سكايب ذو ملكية خاصة (proprietary) فهو حالياً لا يوضّح كيف ينظم الفهرس عبر النظائر الممتازة، بالرغم من أن نوعاً ما من تنظيمات DHT محتملة جداً.

تُستخدَم تقنيات النظائر أيضاً في مرحّلات سكايب (Skype Relays) لعمل مكالمات (calls) بين المضيفات في الشبكات المنزلية. توفر العديد من ترتيبات الشبكات المنزلية اتصالاً بالإنترنت من خلال "موجّه" (عادة يكون موجّهاً لاسلكياً). وهذه "الموجّهات" في الحقيقة أكثر من كونها "موجّهات"، فهي تتضمن عادة ما يطلق عليه مترجم عناوين الشبكة (Network Address (NAT) Translator). وسوف ندرس NAT بالتفصيل في الفصل الرابع. كل ما نحتاج لمعرفته هنا هو أن NAT يمنع مضيفاً من خارج الشبكة المنزلية من بدء الاتصال مع مضيف ضمن الشبكة المنزلية. إذا كان كلا الشخصين المتصلين من خلال سكايب لديهما NAT، فهناك مشكلة حيث لا يستطيع أحدهما تلبية نداء بدأه الشخص الآخر، وبالتالي يبدو الاتصال مستحيلاً. إلا أن الاستعمال الذكي للنظائر الممتازة (super peers) والمُرحِّلات (relays) يحل هذه المشكلة بشكلٍ رائع. افترض أنه عندما تبدأ أليس الاتصال تُنسَب لأحد النظائر الممتازة خارج NAT. تستطيع أليس أن تبدأ جلسة مع نظيرها الممتاز، لأن مترجم عنوان شبكتها NAT يرفض الجلسات التي تبدأ من خارج شبكة منزلها، وهذا يسمح لها ولنظيرها الممتاز بتبادل "رسائل تحكم" على هذه الجلسة. يحدث نفس الشيء لدى بوب. الآن وعندما تريد أليس محادثة بوب، تُخطِر نظيرها الممتاز، والذي يُخطِر بدوره نظير بوب الممتاز، والذي يقوم بعد ذلك بإخطار بوب بالدعوة القادمة من أليس. إذا قَبِل بوب الدعوة لمحادثة أليس، يختار النظيران الممتازان نظيراً ممتازاً ثالثاً خارج NAT (يطلق عليه المرحِّل) تكون وظيفته نقل البيانات بين أليس وبوب. عندئذ يُخطِر النظيران الممتازان أليس وبوب لبدء الاتصال عبر المرحِّل. ترسل أليس حزم بيانات الصوت إلى المرحِّل على التوصيلة Alice-to-relay (والتي أنشئت من

قِبَل أليس)، ومن ثم يُرسِل المرحِّل تلك الرزم على التوصيلة relay-to-Bob (والتي أنشئت من قِبَل بوب) إلى بوب. تتدفق الرزم من بوب إلى أليس في الاتجاه المعاكس على نفس توصيلتي الترحيل هاتين. وبهذه الطريقة ينجح بوب وأليس في الحصول على توصيلة حسب الطلب من طرف إلى طرف رغم أنه لا يمكن لأحدهما أن يقبل جلسة يتم إنشاؤها من خارج شبكة الاتصالات المحلية LAN الخاصة به. يوضح استخدام المُرحِّلات التصميم المتطوّر جداً لأنظمة النظائر، حيث تؤدّي النظائر خدمات نظام رئيسة للآخرين (كخدمات الفهرسة والترحيل)، بينما تستخدم هي نفسها في الوقت ذاته خدمات المُستخدِم النهائي (كتحميل الملفات وهاتف الإنترنت) والمتاحة من نظام النظائر.

يعتبر سكايب تطبيق إنترنت ناجح وواسع الانتشار جداً، حيث امتدت خدماته لتشمل عشرات الملايين من المُستخدِمين. إن التبني السريع والواسع الانتشار بشكلٍ مدهش لسكايب، بالإضافة إلى مشاركة النظائر للملفات والويب والمراسلة الفورية قبل ذلك، لشاهد على حكمة التصميم المعماري العام للإنترنت. ذلك التصميم الذي لم يكن له أن يتوقع تطبيقات الإنترنت الغنية والدائمة التوسع التي تم تطويرها على مدى السنوات الثلاثين التالية لظهوره. إن الخدمات التي توفرها الشبكة لتطبيقات الإنترنت (كالنقل اللاتوصيلي لوحدات البيانات (من خلال UDP)، والنقل التوصيلي الموثوق لوحدات البيانات (من خلال TCP)، وواجهة برمجة المقابس (socket programming interface)، والعنونة (addressing)، ودليل أسماء النطاقات (DNS)، وغيرها من الخدمات الأخرى) قد أثبتت أنها كافية للسماح بتطوير الآلاف من التطبيقات. ونظراً لأن كل تلك التطبيقات تعمل فوق الطبقات الأربع السفلى الحالية من رصّة بروتوكولات الإنترنت، فإنها تتطلب فقط تطوير برامج "زبون/خادم" أو "نظير لنظير" جديدة تستخدم على الأنظمة الطرفية، مما ساعد على انتشار تلك التطبيقات وتبنيها بسرعة.

7-2 برمجة مقابس بروتوكول TCP

الآن وبعد أن استعرضنا عدداً من تطبيقات الشبكة الهامة، دعنا نستكشف كيف تُكتَب برامج تطبيقات الشبكة في الواقع. في هذا الجزء سنكتب برامج تطبيقات تستخدم بروتوكول TCP؛ وفي الجزء التالي سنكتب برامج تطبيقات تستخدم بروتوكول UDP.

تذكّر من الجزء 2-1 أن العديد من تطبيقات الشبكة تتكون من زوج من البرامج – برنامج زبون وبرنامج خادم – موجودين على نظامين طرفيين مختلفين. وعند تشغيل هذين البرنامجين، يتم إنشاء عملية زبون وعملية خادم، وتتصل هاتان العمليتان فيما بينهما عن طريق الكتابة إلى المقابس والقراءة منها. عند تطوير تطبيق للشبكة تتلخص المهمّة الرئيسة لمطوّر التطبيق في كتابة الكود لبرنامجي الخادم والزبون .

يوجد نوعان من تطبيقات الشبكة. نوع يطبق معيار بروتوكول (قياسي) مُعرَّف على سبيل المثال في طلبات التعليقات (RFCs). ولهذا التطبيق يجب أن يتوافق كلٌّ من برنامجي الخادم والزبون مع القواعد التي يمليها طلب التعليقات. على سبيل المثال يمكن أن يكون برنامج الزبون تحقيقاً لجانب الزبون لبروتوكول FTP (والذي تناولناه في الجزء 2-3 والمُعرّف بشكلٍ واضح في طلب التعليقات RFC 959)، وبنفس الطريقة يمكن أن يكون برنامج الخادم تحقيقاً لجانب خادم FTP (والمُعرّف أيضاً بشكلٍ واضح في طلب التعليقات RFC 959). وإذا كتب أحد مطوّري التطبيقات الكود لبرنامج الزبون، وكتب مطوّر آخر مستقل الكود لبرنامج الخادم، واتبع كلا المطوّرين قواعد RFC بعناية، فسوف يكون البرنامجان قادرين على التعامل مع بعضهما البعض. وفي الحقيقة يتضمن العديد من تطبيقات الشبكة اليوم اتصالاً بين برامج الزبون والخادم والتي طورت من قِبَل مطوّرين مستقلين. على سبيل المثال يتصل متصفّح Firefox بخادم ويب Apache، أو يُحمِّل برنامج زبون FTP (موجود على جهاز حاسب شخصي) ملفاً إلى خادم لاينكس FTP. عندما يطبق زبون أو خادم بروتوكولاً مُعرّفاً في RFC، يجب

استخدام رقم المنفذ المقترن بالبروتوكول. (نوقشت أرقام المنافذ باختصار في الجزء 2-1، وسوف تُغطى بتفصيل أكثر في الفصل الثالث). النوع الآخر لتطبيق الشبكة هو تطبيق ذو ملكية خاصة (proprietary)، وفي هذه الحالة لا يتوافق بروتوكول طبقة التطبيقات المستخدم من قِبَل برامج الخادم والزبون بالضرورة مع طلبات تعليقات موجودة حالياً. عند تطوير كلا البرنامجين للخادم والزبون يكون للمطوّر سيطرة كاملة على ما يكتب في الكود. ولكن نظراً لأن الكود لا يُطبّق بروتوكول ذا ملكية عامة (public-domain protocol)، فلن يستطيع مطوّرون مستقلون آخرون تطوير كود آخر يمكنه التعامل مع ذلك التطبيق. وعند تطوير تطبيق ذي ملكية خاصة يجب أن يتوخى المطوّر الحذر وألا يستخدم أحد أرقام المنافذ المشهورة والمُعرّفة في طلبات التعليقات.

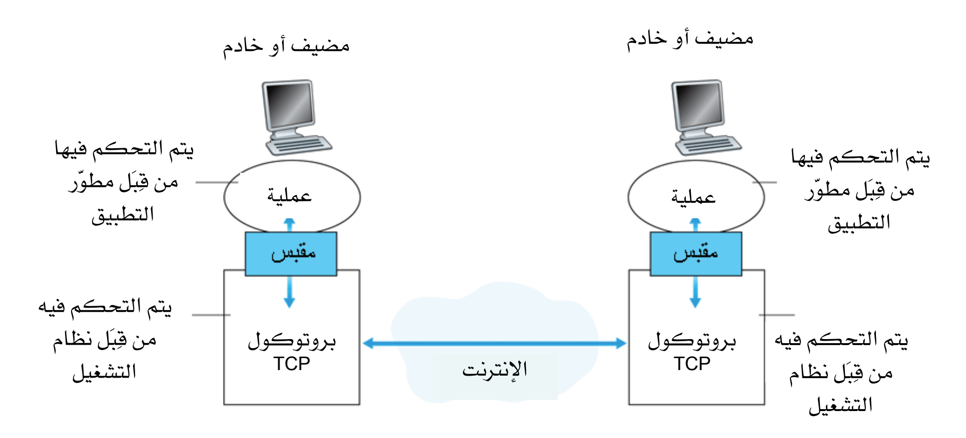
في هذا الجزء والجزء التالي سوف نتناول القضايا الرئيسة في تطوير تطبيق "زبون/خادم" خاص. من أول القرارات التي يجب أن يتخذها المطوّر هو "هل يعمل التطبيق على بروتوكول TCP أو على بروتوكول UDP؟". تذكّر أن بروتوكول TCP توصيلي ويوفر قناة مجرى للتدفق الموثوق لبايتات البيانات (reliable byte-stream channel) بين نظامين طرفيين. أما بروتوكول UDP فغير توصيلي ويرسل رزم بيانات مستقلة من نظام طرفي إلى آخر بدون أي ضمانات فيما يتعلق بالتوصيل.

في هذا الجزء سنُطوِّر تطبيق زبون بسيط يعمل على بروتوكول TCP، وفي الجزء التالي سنُطوّر برنامج زبون بسيط يعمل على بروتوكول UDP. وسوف نكتب هذه البرامج البسيطة بلغة البرمجة جافا. رغم أنه كان من الممكن أن نكتب الكود بلغة C أو لغة C++، لكننا اخترنا لغة جافا لأن التطبيقات تكتب فيها على نحو نظيف وبشكلٍ أكثر تنسيقاً. فمع لغة جافا توجد سطور أقل في كود البرنامج، وكل سطر يسهل توضيحه للمبرمج المبتدئ بدون صعوبة كبيرة. لكن ليس هناك حاجة لكي تخاف إذا كنت غير ملم بلغة جافا. ستصبح قادراً على تتبع كود جافا إذا كان لديك خبرة في البرمجة بلغة أخرى. وتوجد عدة

مراجع جيدة للقرّاء المهتمّين ببرمجة تطبيقات زبون/خادم بلغة C ‏[Donahoo 2001; Stevens 1997; Frost 1994; Kurose 1996].

2-7-1 برمجة مقابس TCP

ذكرنا في الجزء 2-1 أن العمليات تنفذ على حاسبات مختلفة تتصل فيما بينها بإرسال الرسائل إلى المقابس. وقلنا أن كل عملية تناظر البيت ومقبس العملية يناظر باب البيت. وكما هو موضح في الشكل 2-30 يُعتبر المقبس بمثابة الباب بين العملية وبروتوكول TCP. ولمطوّر البرنامج سيطرة كاملة على كل شيء على جانب طبقة التطبيقات من المقبس، ولكن ليس له سيطرة تذكر على جانب طبقة النقل من المقبس (فعلى الأكثر يكون له القدرة على ضبط بضعة بارامترات لبروتوكول TCP كالحجم الأقصى للذاكرة المؤقتة (buffer) والحجم الأقصى لقطعة بيانات TCP).



الشكل 2-30 اتصال العمليات عن طريق مقابس TCP.

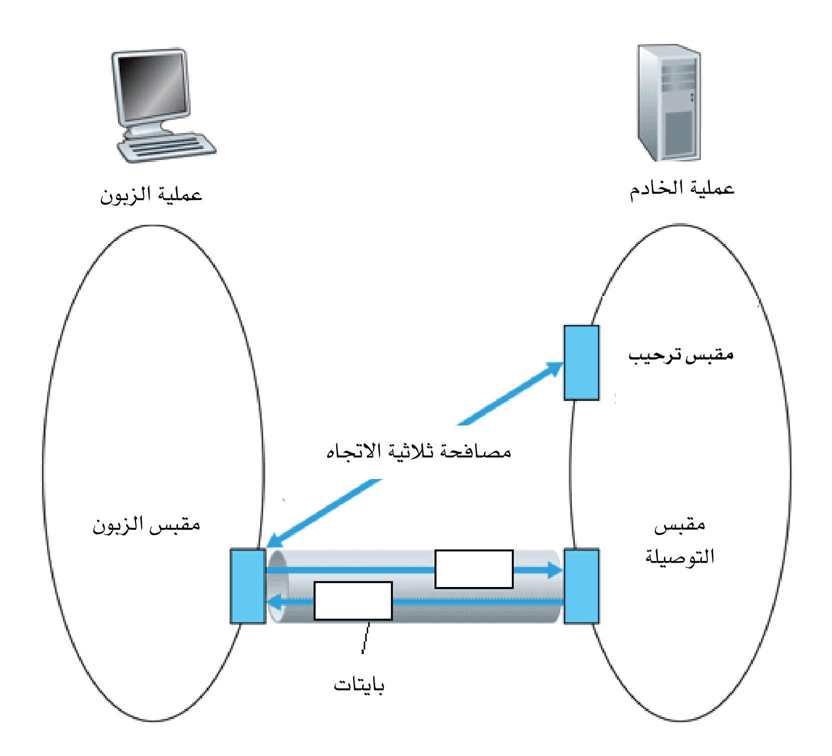
دعنا الآن نلقي نظرة أدق على التفاعل ما بين برنامجي الخادم والزبون. يتحمل الزبون مسؤولية بدء الاتصال بالخادم، ولكي يتمكّن الخادم من الرد على اتصالِ الزبون الأولي، يجب أن يكون الخادم جاهزاً، وهذا يتطلب شيئين. أولاً: لا

يمكن أن يكون برنامج الخادم خاملاً (dormant)، فعليه أن يكون شغّالاً قبل أن تحاول عملية الزبون بدء الاتصال. ثانياً: يجب أن يكون لدى برنامج الخادم نوع من الباب (وبدقة أكثر مقبس) يُرحّب ببعض الاتصال الأولي من عملية زبون تنفّذ على مضيف اعتباطي. باستخدام التناظر بين "بيت ↔ عملية" و"باب ↔ مقبس" سنشير أحياناً إلى اتصال الزبون الأولي على أنه "طرْق على باب الترحيب".

عندما تكون عملية الخادم شغّالة، يمكن أن تبدأ عملية الزبون توصيلة TCP إلى الخادم. وهذا يتم في برنامج الزبون بإنشاء مقبس. عندما يُنشئ زبونٌ ما مقبساً، يحدد عنوان عملية الخادم الذي سيتم الاتصال به (والذي يتضمن عنوان IP لمضيف الخادم ورقم منفذ عملية الخادم). وبمجرد إنشاء المقبس في برنامج الزبون، يقوم TCP في الزبون ببدء مصافحة ثلاثية (3-way handshaking) ويُنشئ توصيلة TCP إلى الخادم. إن المصافحة الثلاثية التي تحدث في طبقة النقل غير مرئية تماماً لبرامج الخادم والزبون.

أثناء المصافحة الثلاثية تطرق عملية الزبون على باب الترحيب لعملية الخادم. وعندما "يسمع" الخادم الطرْق، يُنشئ باباً جديداً (بالأحرى مقبساً جديداً) يُكرَّس لذلك الزبون. في المثال الذي سنتناوله باب الترحيب هو كائن من نوع ServerSoeket ونطلق عليه welcomeSocket. وعندما يطرق زبونٌ ما على هذا الباب، يستدعي البرنامج الوظيفة welcomeSocket.accept()، والتي تنشئ باباً جديداً للزبون. في نهاية مرحلة المصافحة تؤسس توصيلة TCP بين مقبس الزبون ومقبس الخادم الجديد. ومن الآن فصاعداً سنشير إلى مقبس الخادم الجديد بمقبس توصيلة الخادم (connection socket). وهذه المقابس موضحة في الشكل 2-31.

من منظور التطبيق تمثل توصيلة TCP أنبوباً افتراضياً بين مقبس الزبون ومقبس توصيلة الخادم. يمكن أن ترسل عملية الزبون بايتات اعتباطيةَ إلى مقبسها، وسوف يضمن TCP توصيل كل بايت أُرسِل في الطلب إلى عملية الخادم (خلال مقبس التوصيل) بنفس ترتيب إرساله. وهكذا يوفر TCP خدمة مجرى



الشكل 2-31 ثلاث مقابس: مقبس الزبون، ومقبس الترحيب، ومقبس التوصيلة.

للتدفق الموثوق لبايتات البيانات بين عمليتي الخادم والزبون. علاوة على ذلك (وكما يمكن للناس الدخول والخروج من نفس الباب) فإن عملية الزبون لا ترسل البايتات فقط إلى مقبسها، ولكنها تستلم البايتات أيضاً من نفس المقبس. بنفس الطريقة فإن عملية الخادم لا تستلم البايتات فقط من مقبس توصيلتها، ولكنها ترسل البايتات أيضاً عبر ذلك المقبس. ولما كانت المقابس تلعب دوراً محورياً في تطوير تطبيقات زبون/خادم فإن عملية التطوير تُعرَف ببرمجة المقابس (socket programming).

قبل إعطاء مثال لتطبيق زبون/خادم، من المفيد مناقشة المصطلح "مجرى" (stream)، فهو عبارة عن سيل من الحروف يتدفق داخل أو خارج عملية. وكل "مجرى" إما أن يكون "مجرى إدخال" (input stream) أو "مجرى إخراج" (output

stream) للعملية. يرتبط مجرى الإدخال بوحدة إدخال للعملية كوحدة الإدخال القياسية (لوحة المفاتيح (keyboard)) أو كمقبس تتدفق البيانات عبره من الإنترنت، في حين يرتبط مجرى الإخراج بوحدة إخراج للعملية كوحدة الإخراج القياسية (شاشة العرض (monitor)) أوكمقبس تتدفق البيانات عبره إلى الإنترنت.

2-7-2 مثال لتطبيق "زبون/خادم" بلغة جافا

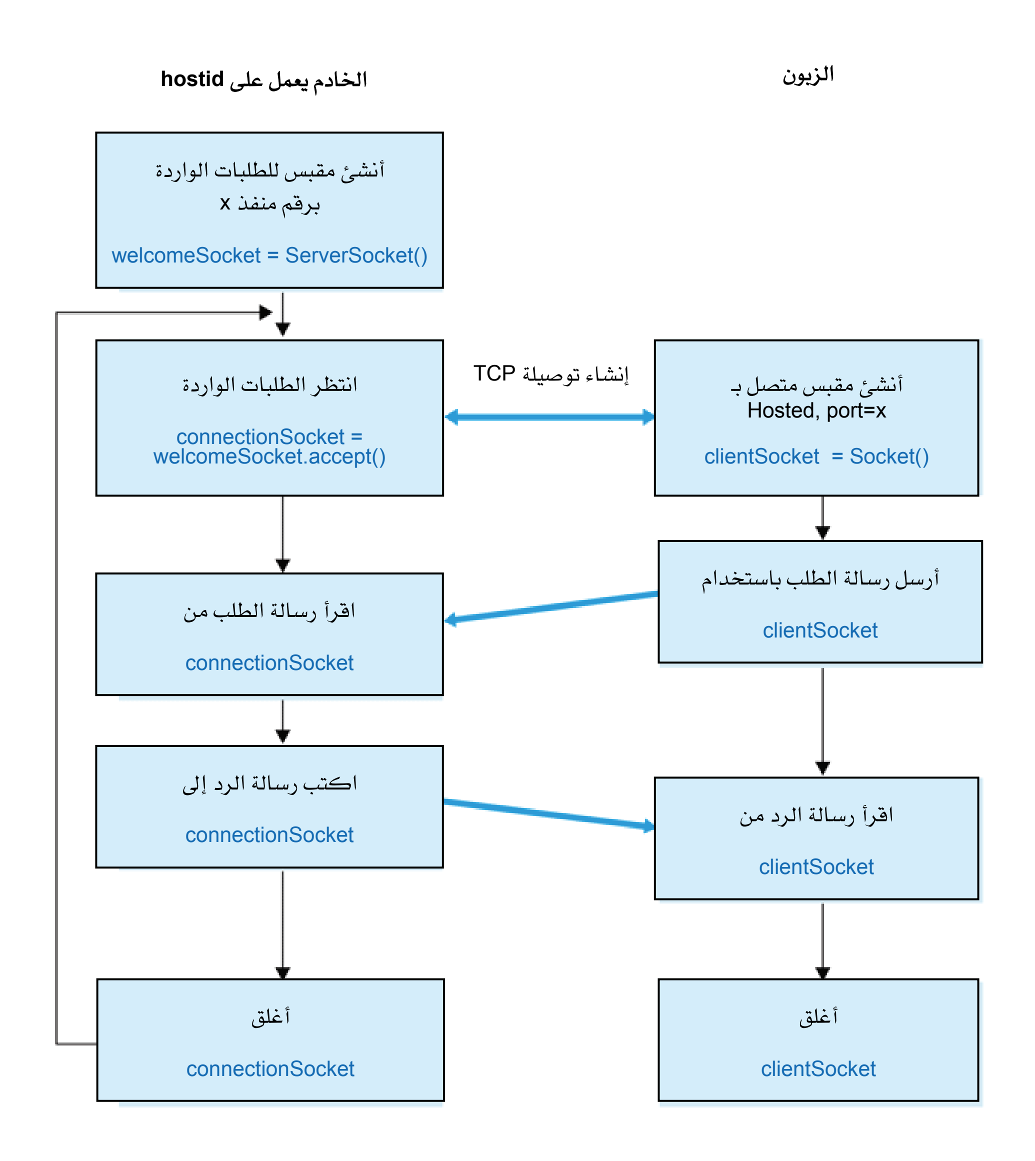
سنستخدم تطبيق زبون/خادم البسيط التالي لتوضيح برمجة المقابس لكلٍّ من TCP وUDP:

1. يقرأ الزبون سطراً من وحدة الإدخال القياسية (لوحة المفاتيح) ويرسله خارج مقبسه إلى الخادم .
2. يقرأ الخادم السطر من مقبس توصيلته مع الزبون.
3. يُحوِّل الخادم حروف السطر إلى حروف كبيرة (uppercase).
4. يُرسِل الخادم السطر المُعدَّل خارج مقبس توصيلته إلى الزبون.
5. يقرأ الزبون السطر المُعدَّل من مقبسه ويعرضه على وحدة الإخراج القياسية (شاشة العرض).

يبين الشكل 2-32 الأنشطة الرئيسة المتعلقة بالمقبس لدى كلٍّ من الزبون والخادم.

نورد فيما يلي زوج البرامج للزبون والخادم مبني على بروتوكول TCP، مع تحليل مفصّل (سطر بسطر) لكل برنامج. يسمى برنامج الزبون TCPClient.java، ويسمى برنامج الخادم TCPServer.java. ولكي نؤكّد على القضايا الرئيسة، سنعطي عمداً كوداً واضحاً للغاية ويفي بالغرض لكنه غير مثالي بالضرورة. سوف يحتوي "الكود الجيد" بالتأكيد على بضعة سطور إضافية للمساعدة. وبمجرد إتمام عملية الترجمة (compilation) للبرنامجين على مضيفاتهم الخاصة، يُنفَّذ برنامج الخادم أولاً في مضيف الخادم ليُنشئ عملية خادم. كما ناقشنا سابقاً تنتظر عملية الخادم لكي تتصل بها عملية زبون. في هذا المثال عندما يُنفّذ برنامج الزبون يتم إنشاء عملية في مضيف الزبون، وهذه العملية تتصل بالخادم فوراً

وتنشئ توصيلة TCP معه. بعد ذلك يمكن للمُستخدِم عند الزبون استخدام التطبيق لإرسال سطر وبعد ذلك يتلقى نسخة من السطر المعدّل من الخادم.



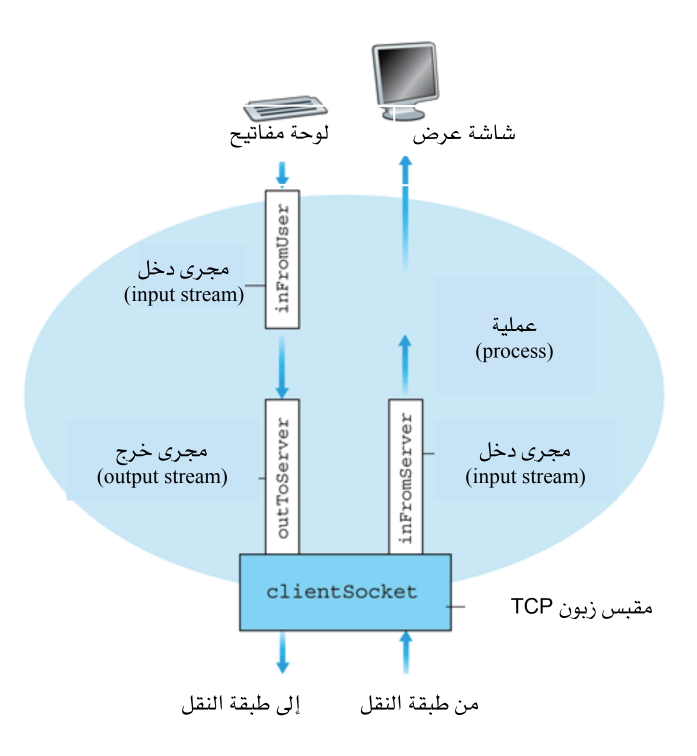
الشكل 2-32 تطبيق "زبون/خادم" باستخدام خدمة النقل التوصيلية (TCP).

برنامج الزبون TCPClient.java

فيما يلي الكود الخاص بجانب الزبون من التطبيق:

```
import java.io.*;
import java.net.*;
class TCPClient {
    public static void main(String argv[]) throws Exception {
        String sentence;
        String modifiedSentence;
        BufferedReader inFromUser = new new BufferedReader(new
        InputStreamReader(System.in));
        Socket clientSocket = new Socket("hostname", 6789);
        DataOutputStream outToServer = new DataOutputStream(
        clientSocket.getOutputStream());
        BufferedReader inFromServer = new BufferedReader(new
        InputStreamReader(clientSocket.getInputStream()));
        sentence = inFromUser.readLine();
        outToServer.writeBytes(sentence+'\n');
        modifiedSentence = inFromServer.readLine();
        System.out.println("FROM SERVER: "+modifiedSentence);
        clientSocket.close();
    }
}
```

يُنشئ برنامج TCPClient.java مقبساً واحداً وثلاثة مجارٍ (streams) كما هو موضح في الشكل 2-33. يُسمى المقبس clientSocket. يُستخدَم المجرى inFromUser لمدخلات البرنامج ويرتبط بوحدة الإدخال القياسية (لوحة المفاتيح).عندما يكتب المُستخدِم حروفاً على لوحة المفاتيح، تتدفق الحروف من خلال المجرى inFromUser. أمّا المجرى inFromServer فهو مجرى آخر للمدخلات إلى البرنامج مرتبط بالمقبس. تصب الحروف التي تصل من شبكة الإنترنت في مجرى inFromServer. أخيراً يُستخدم المجرى outToServer لمخرجات برنامج الزبون إلى الخادم وهو مرتبط بالمقبس أيضاً. تتدفق الحروف التي يرسلها الزبون إلى الشبكة في المجرى outToServer.



الشكل 2-33 يمتلك الزبون TCPCleint ثلاثة مجارٍ تمر خلالها الحروف.

دعنا الآن نلقي نظرة على السطور المختلفة في الكود.

```
import java.io.*;
import java.net.*;
```

تمثل java.io وjava.net حزمتين من حزم لغة جافا. تحتوي حزمة java.io على فصائل (classes) تشمل النوع "مجرى" للإدخال والإخراج. وبشكلٍ خاص تتضمن الحزمة BufferedReader java.io وDataOutputStream التي يستخدمها البرنامج لإنشاء أنواع "المجرى" الثلاثة التي سبق توضيحها. أمّا حزمة java.net فتحتوي على فصائل لدعم الشبكة. وبشكلٍ خاص تحتوي على Socket وServerSocket. يعرّف البرنامج الكائن clientSocket من النوع Socket. أما السطور التالية:

```
class TCPClient{
public static void main(String argv[]) throws Exception{ ...... }
```

فهي جزء قياسي يوجد في مقدمة معظم برامج جافا. تبدأ الكلمة الدليلية "class" تعريف فصيلة تسمى TCPClient. تحتوي الفصيلة على متغيرات (variables) ووظائف (methods). تحدد بداية ونهاية كتلة تعريف الفصيلة بالأقواس {}. الفصيلة TCPClient ليس لها متغيرات وتتضمن وظيفة واحدة فقط تسمى main() تشبه الوظائف في لغة جافا الإجراءات في لغات أخرى كلغة C، كما تشبه الوظيفة main() نظيراتها في لغات C وC++. عندما يُنفِّذ مترجم جافا (Java Interpreter) تطبيقاً (باستدعائه من قِبَل فصيلة التحكم في التطبيق)، يبدأ باستدعاء الوظيفة main() في الفصيلة، والتي تستدعى بدورها الوظائف الأخرى المطلوبة لتنفيذ التطبيق. في هذه المقدمة لبرمجة المقابس في لغة جافا، يمكنك إهمال الكلمات الدليلية

public, static, void, main, throws Exceptions

(رغم ضرورة تضمينها في الكود).

```
String sentence;
String modifiedSentence;
```

يُعرِّف هذان السطران كائنين من نوع سلسلة الحروف (String). يستخدم الكائن الأول sentence لحفظ سلسلة الحروف التي تكتب من قِبَل المُستخدِم والمراد إرسالها إلى الخادم. أما الكائن الثاني modifiedSentence فيُستخدَم لحفظ سلسلة الحروف التي يحصل عليها الزبون من الخادم والتي ترسل لوحدة الإخراج القياسية لدى المُستخدِم.

```
BufferedReader inFromUser = new BufferedReader(new
InputStreamReader(System.in());
```

يُنشئ هذا السطر كائن inFromUser من النوع BufferedReader، وبدايةً يأخذ القيمة System.in والتي تربط "مجرى الإدخال" بوحدة الإدخال القياسية، مما يسمح للزبون بقراءة النص من لوحة المفاتيح لديه.

```
Socket client Socket = new Socket ("hostname", 6789);
```

يُنشئ هذا السطر الكائن clientSocket من النوع Socket، كما يبدأ أيضاً توصيلة TCP بين الزبون والخادم. ويجب أن تستبدل سلسلة الحروف "hostname" باسم مضيف الخادم الحقيقي (على سبيل المثال"apple.poly.edu"). وقبل أن يبدأ تشغيل توصيلة TCP في الواقع، يقوم الزبون بالحصول على عنوان IP المناظر لاسم المضيف عن طريق DNS. يمثل العدد 6789 رقم منفذ الخادم، ويمكنك استعمال رقم منفذ مختلف، لكن يجب أن تتأكد من أنك تستعمل نفس رقم المنفذ في جانب الخادم من التطبيق. كما ذكرنا في السابق، يستخدم عنوان IP للمضيف مع رقم منفذ التطبيق في تمييز عملية الخادم .

```
DataOutputStream outToServer  = new
DataOutputStream(clientSocket.getOutputStream());
BufferedReader inFromServer= new BufferedReader(new
inputStreamReader(clientSocket.getInputStream()));
```

يُنشئ هذان السطران كائنات مجرى ترتبط بالمقبس. يُستخدَم outToServer لإرسال ناتج العملية إلى الخادم، بينما يُستخدَم inFromServer للاستقبال من الخادم (انظر الشكل 2-33).

```
Sentence = inFromUser.readLine();
```

يضع هذا السطر ما كتبه المُستخدِم على لوحة المفاتيح في الكائن sentence. ويستمر هذا الكائن في استقبال الحروف التي يكتبها المُستخدِم إلى أن ينهي المُستخدِم الكتابة بالضغط على رمز "بداية السطر". تمر تلك الحروف من وحدة الإدخال القياسية (لوحة المفاتيح) خلال المجرى inFromUser إلى الكائن sentence.

```
outToServer.writeBytes(sentence + '\n');
```

يُرسِل هذا السطر النص الموجود بالكائن sentence مع رمز "بداية السطر" إلى المجرى outToServer. تتدفق الحروف خلال مقبس الزبون إلى أنبوب TCP ومن ثم إلى الخادم .

```
modifiedSentence = inFromServer.readLine();
```

يسبب هذا السطر انتظار عملية الزبون لحين تلقي رسالة من الخادم. عندما تصل الحروف من الخادم، تتدفق خلال المجرى inFromServer وتوضع في الكائن modifiedSentence. يستمر تجميع الحروف في modifiedSentence حتى ينتهي السطر برمز "بداية السطر".

```
System.out.println("FROM SERVER" + modifiedSentence);
```

يعرض هذا السطر محتوى الكائن modifiedSentence على الشاشة.

```
clientSocket.close();
```

يغلق هذا السطر الأخير المقبس، وبالتالي يغلق توصيلة TCP بين الزبون والخادم، مما يجعل TCP في الزبون يرسل رسالة TCP إلى TCP في الخادم لإنهاء التوصيلة (انظر الجزء 3-5).

برنامج الخادم TCPServer.java

لنلقِ الآن نظرة على برنامج الخادم. يشبه برنامج الخادم TCPServer.java برنامج الزبون TCPClient.java في عدة جوانب. دعنا الآن نلقي نظرة على سطور البرنامج TCPServer.java، مع ملاحظة أننا لن نُعلِّق على السطور المطابقة أو المشابهة للأوامر في برنامج الزبون TCPClient.java.

يختلف السطر الأول في TCPServer.java بشكلٍ جوهري عن نظيره في TCPClient.java:

```
ServerSocket welcomeSocket = new ServerSocket(6789);
```

يُنشئ هذا السطر الكائن welcomeSocket من نوع ServerSocket، وهو بمثابة باب يستمع الخادم للطرق عليه من بعض الزبائن، وهو معرّف برقم المنفذ 6789.

```
import java.io.*;
import java.net.*;
class TCPServer {
    public static void main(String argv[]) throws Exception{
        String clientSentence;
        String capitalizedSentence;
        ServerSocket welcomeSocket = new ServerSocket(6789);
        while(true){
            Socket connectionSocket = welcomeSocket.accept();
            BufferedReader inFromSocket = new BufferedReader(new
            InputStreamReader(connectionSocket.getInputStream()));
            DataOutputStream outToClient = new DataOutputStream(
            connectionSocket.getOutputStream());
            clientSentence = inFromClient.readLine();
            capitalizedSentence = clientSentence.toUpperCase() + '\n';
            outToClient.writeBytes(capitalizedSentence);
        }
    }
}
```

Socket connectionSocket = welcomeSocket.accept();

يُنشئ هذا السطر مقبساً جديداً يسمّى connectionSocket عندما يطرق زبون ما على welcomeSocket. يستخدم المقبس منفذ رقم 6789 أيضاً (وسوف نوضّح في الفصل الثالث لماذا يكون لكلا المقبسين نفس رقم المنفذ). بعد ذلك يُنشئ TCP أنبوباً افتراضياً بين clientSocket في الزبون وconnectionSocket في الخادم. عندئذ يمكن أن يرسل كلٌّ من الزبون والخادم البايتات إلى بعضهم البعض عبر هذا الأنبوب، وتصل كل البايتات المُرسَلة إلى الجانب الآخر بنفس الترتيب. بعد إنشاء connectionSocket يمكن للخادم أن يواصل الاستماع للطلبات من عملاء التطبيق الآخرين باستعمال welcomeSocket، إلا أن هذه النسخة من البرنامج لا تستمع في الحقيقة لمزيد من طلبات الاتصال ولكن يمكن تعديل البرنامج باستخدام threads

للقيام بذلك. بعد ذلك يُنشئ البرنامج عدة كائنات من نوع "مجرى" مماثلة لتلك الكائنات التي تم إنشاؤها في clientSocket.

```
capitalizedSentence = clientSentence.toUpperCase() + '\n';
```

يمثل هذا الأمر قلب التطبيق، فهو يأخذ السطر (سلسلة الحروف) الذي أرسل من قِبَل الزبون، ويحول الحروف إلى حروف كبيرة. باستعمال الوظيفة toUpperCase()، ثم يضيف رمز "بداية السطر". تعتبر بقية الأوامر الأخرى في البرنامج ثانوية فهي تُستخدَم أساساً للاتصال مع الزبون .لاختبار زوج البرامج قم بإنشاء الملف TCPClient.java على مضيف والملف TCPServer.java على مضيف آخر. تأكّد من تضمين اسم المضيف الصحيح للخادم في برنامج الزبون TCPClient.java، ثم قم بترجمة كلا البرنامجين لتحصل على الملفين المناظرين TCPServer.class وTCPClient.class القابلين للتنفيذ. نفّذ البرنامج TCPServer.class على الخادم فتتكوّن عملية خادم تبقى خاملة إلى أن يتصل بها زبون ما. ثم نفذ البرنامج TCPClient.class فتتكوّن عملية زبون يتم إنشاء توصيلة TCP بين عمليتي الخادم والزبون. أخيراً لاستعمال التطبيق اكتب جملة تليها علامة "بداية السطر" (بالضغط على مفتاح الإدخال) على الزبون.

لتطوير تطبيقات زبون/خادم خاصة بك يمكنك أن تبدأ بإدخال بعض التعديلات على البرنامجين السابقين. على سبيل المثال بدلاً من أن يُحوّل برنامج الخادم كل الحروف إلى حروف كبيرة، يمكن أن يحسب عدد مرات ظهور الحرف "s" في الرسالة ثم يُرجع ذلك العدد.

8-2 برمجة مقابس بروتوكول UDP

عرفنا في الجزء السابق أنه عندما تتصل عمليتان على بروتوكول TCP فكأننا قد مدّدنا أنبوباً لنقل البايتات بينهما، ويبقى مفعول هذا الأنبوب سارياً حتى تغلقه إحدى العمليتين. عندما تريد إحدى العمليات إرسال بعض البايتات إلى العملية الأخرى، فإنها ببساطة تقوم بإدخال البايتات إلى الأنبوب. لا يتعين على عملية الإرسال أن تربط عنوان "الوجهة النهائية" بالبايتات؛ حيث أن الأنبوب متصل

منطقياً بالوجهة النهائية. وعلاوة على ذلك يوفر الأنبوب قناة لمجرى بايتات يوفر نقلاً موثوقاً للبيانات – أي أن سلسلة البايتات التي تتلقاها عملية الاستقبال لدى المُستقبِل هي بالضبط نفس سلسلة البايتات التي أدخلها المُرسِل إلى الأنبوب.

يسمح بروتوكول UDP أيضاً لعمليتين أو أكثر تعملان على مضيفين مختلفين بالاتصال، ولكنه يختلف عن بروتوكول TCP في العديد من الجوانب الأساسية. أولاً: يوفر بروتوكول UDP خدمة غير توصيلية – فلا توجد مرحلة مصافحة في البداية ولا يتم إنشاء أنبوب اتصال بين العمليتين. ولذا فعندما تريد عملية إرسال دفعة من البايتات إلى العملية الأخرى، يجب أن تربط عملية "المُرسِل" عنوان العملية المقصودة في الوجهة النهائية بدفعة البايتات تلك. ويجب أن يتم ذلك لكل دفعة من البايتات ترسلها عملية المُرسِل. وكمثال لنأخذ بعين الاعتبار مجموعة من عشرين شخصاً تستقل خمس سيارات أجرة إلى وجهة نهائية مشتركة؛ بينما يدخل الأشخاص سيارات الأجرة التي ستقلهم، يجب أن يُخطَر كل سائق سيارة أجرة على حدة بالوجهة المطلوبة. يشبه نموذج خدمة UDP سيارة الأجرة تلك. يشمل عنوان الوجهة النهائية عنوان IP لمضيف الوجهة ورقم منفذ العملية على ذلك المضيف. يطلق على دفعة بايتات المعلومات مع عنوان IP لمضيف الوجهة ورقم المنفذ "رزمة" بيانات. يوفر UDP نموذج خدمة مبنياً على الرسائل (message-oriented)، وهذا يعني أن البايتات التي ترسل كدفعة واحدة على جانب الإرسال ويتم توصيلها معاً إلى جانب الاستلام. وهذا يختلف عن طريقة TCP الذي يرسل سيلاً من البايتات المنفصلة في مجرى توصيل موثوق. تُعتبر خدمة UDP خدمة من نوع "أفضل جُهد"، فلا يوجد في الواقع أي ضمان لتوصيل دفعة البايتات المُرسَلة. وهكذا تختلف خدمة UDP بشدة (في عدة نواحٍ) عن نموذج خدمة TCP لمجرى البايتات الذي يحقق نقلاً موثوقاً للبيانات.

بعد تكوين "رزمة" البيانات، تدفع عملية الإرسال بالرزمة إلى الشبكة عبر مقبس. واستمراراً مع مثال سيارة الأجرة، توجد على الجانب الآخر من مقبس الإرسال سيارة أجرة تنتظر الرزمة. تحمل سيارة الأجرة الرزمة عندئذ في اتجاه عنوان الوجهة النهائية للرزمة. ومع ذلك لا تضمن سيارة الأجرة توصيل الرزمة في

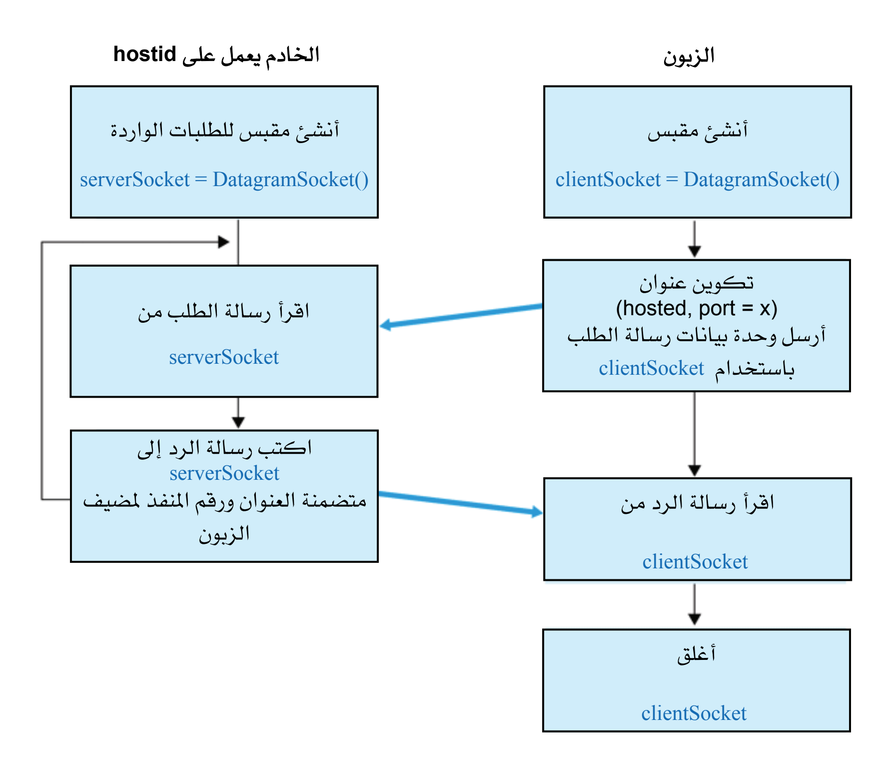
النهاية إلى وجهتها النهائية، حيث يمكن أن تتعطل سيارة الأجرة أو تعاني من مشكلة أخرى غير متوقعة. وبمعنى آخر يوفر بروتوكول UDP خدمة نقل غير موثوقة للعمليات التي تستخدمه للاتصال فيما بينها، أي بدون ضمانات لتوصيل الرزمة إلى غايتها النهائية.

سنتناول في هذا الجزء برمجة المقابس بإعادة تطوير نفس التطبيق في الجزء السابق، ولكن هذه المرة على UDP. سوف نرى أن الكود لبروتوكول UDP مختلف عن الكود لبروتوكول TCP في العديد من الجوانب المهمة. وعلى وجه التحديد: (1) لا توجد مصافحة في البداية بين العمليتين ولذا فلا حاجة لمقبس ترحيب، (2) لا توجد كائنات "مجرى" مرتبطة بالمقبس، (3) تكوّن مضيفات الإرسال الرزم بربط عنوان الوجهة النهائية ورقم المنفذ مع كل دفعة بايتات يتم إرسالها، (4) يجب أن تقوم عملية الاستقبال بفض كل رزمة يتم استلامها للحصول على بايتات المعلومات الموجودة داخل الرزمة.

تذكّر مرة أخرى تطبيقنا البسيط:

1. يقرأ الزبون سطراً من وحدة الإدخال القياسية (لوحة المفاتيح) ويرسله خارج مقبسه إلى الخادم.
2. يقرأ الخادم السطر من مقبسه.
3. يُحوّل الخادم حروف السطر إلى حروف كبيرة (uppercase).
4. يُرسِل الخادم السطر المُعدَّل خارج مقبسه إلى الزبون .
5. يقرأ الزبون السطر المُعدَّل من مقبسه ويعرضه على وحدة الإخراج القياسية (شاشة العرض).

يوضح الشكل 2-34 الأنشطة الرئيسة بين الزبون والخادم اللذين يتصلان بخدمة نقل غير توصيلية (UDP).



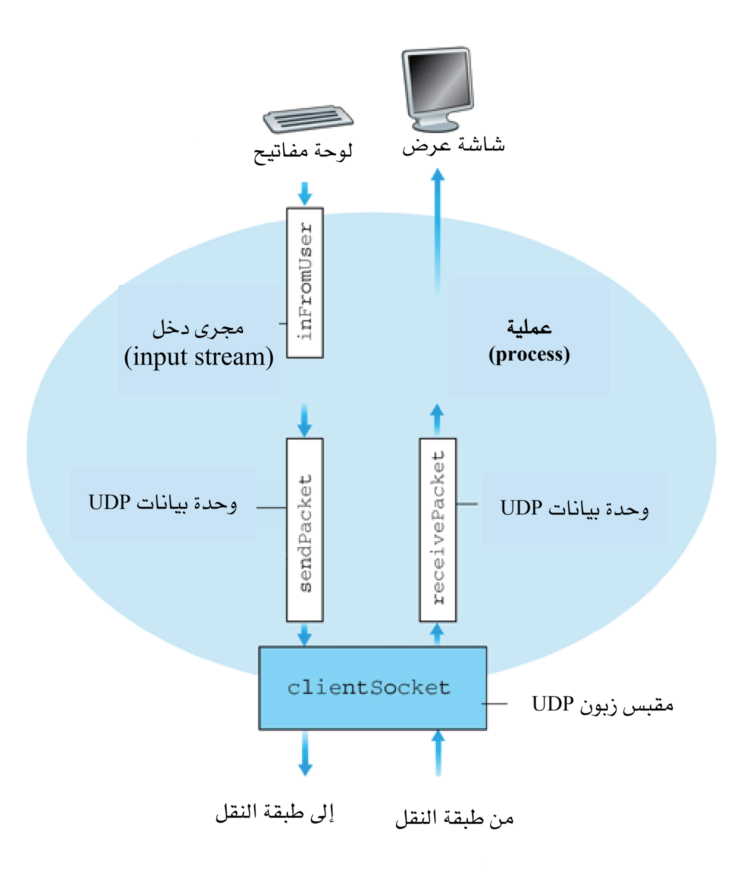
الشكل 2-34 تطبيق "زبون/خادم" باستخدام خدمات نقل غير توصيلية (UDP).

برنامج الزبون UDPClient.java

يُنشئ البرنامج UDPClient.java الخاص بجانب الزبون من التطبيق مجرى واحداً ومقبساً واحداً، كما هو موضح في الشكل 2-35. يُدعى المقبس clientSocket وهو من نوع DatagramSocket. لاحظ أن UDP في الزبون يستخدم نوعاً مختلفاً من المقابس عن TCP. بالتحديد مع UDP يستخدم الزبون مقبساً من نوع DatagramSocket، بينما يستخدم زبون TCP مقبساً من نوع Socket. المجرى inFromUser هو مجرى إدخال إلى البرنامج ويرتبط بوحدة الإدخال القياسية (لوحة المفاتيح) (كان لدينا "مجرى" مكافئ في نسخة TCP من البرنامج). ولكن بالمقارنة مع TCP لايوجد مجرى (إدخال أو إخراج) مرتبط بالمقبس. يدفع UDP

رزماً منفصلة خلال كائن من النوع DatagramSocket بدلاً من أن يغذي البايتات إلى مجرى مرتبط بكائن من النوع Socket (كما كان الحال مع TCP).

```
import java.io.*;
import java.net.*;
class UDPClient {
	public static void main(String argv[]) throws Exception {
		BufferedReader inFromUser = new new BufferedReader(new
		InputStreamReader(System.in));
		DatagramSocket clientSocket = new DatagramSocket();
		InetAddress IPAddress = InetAddress.getByName("hostname");
		byte[] sendData = new byte[1024];
		byte[] receiveData = new byte[1024];
		String sentence = inFromUser.readLine();
		sendData = sentence.getBytes();
		DatagramPacket sendPacket = new DatagramPacket(sendData,
		sendData.length, IPAddress, 9876);
		clientSocket.send(sendPacket);
		DatagramPacket receivePacket = new DatagramPacket(
		receiveData, receiveData.length);
		clientSocket.receive(receivePacket);
		String modifiedSentence = new String(receivePacket.getData());
		System.out.println("FROM SERVER: "+modifiedSentence);
		clientSocket.close();
	}
}
```



الشكل 2-35 لدى الزبون UDPClient.java مجرى واحد؛ يستلم المقبس رزماً من العملية ويسلمها رزماً.

دعنا الآن نلقي نظرة على سطور الكود التي تختلف بشكلٍ ملحوظ عن برنامج TCPClient.java.

```
DatagramSocket clientSocket = new DatagramSocket();
```

يُنشئ هذا السطر الكائن clientSocket من النوع DatagramSocket. بالمقارنة مع TCPClient.java لا يُنشئ هذا السطر توصيلة TCP. وبالتحديد فإن مضيف الزبون لا يتصل بمضيف الخادم بعد تنفيذ هذا السطر. ولهذا السبب لا تأخذ الوظيفة DatagramSocket() اسم أو رقم منفذ مضيف الخادم كمعاملات للوظيفة

(arguments). باستخدام التناظر "باب ↔ مقبس"، يؤدي تنفيذ السطر أعلاه إلى إنشاء "باب" لعملية الزبون ولكنه لا يُنشئ "أنبوب" اتصال بين العمليتين.

```
InetAddress IPAddress = InetAddress.getByName("hostname") ;
```

لإرسال البايتات إلى عملية الوجهة النهائية، نحتاج إلى عنوان العملية. يمثل عنوان IP لمضيف الوجهة النهائية جزءاً من هذا العنوان. يستدعي السطر أعلاه بروتوكول DNS الذي يُترجِم اسم المضيف (والمزوّد في هذا المثال من قِبَل مطوّر البرنامج) إلى عنوان IP. في نسخة TCP من برنامج الزبون اُستخدِم DNS أيضاً، غير أن ذلك تم ضمنياً وليس بشكلٍ صريح. تأخذ الوظيفة ()getByName اسم مضيف الخادم كمعامل وترجع عنوان IP لهذا الخادم نفسه، ويوضع هذا العنوان في كائن يسمى IPAddress من النوع InetAddress.

```
byte[] sendData = new byte[1024];
byte[] receiveData = new byte[1024];
```

تستخدم مصفوفتا البايتات sendData وreceiveData في تخزين البيانات التي يرسلها ويستلمها الزبون على التوالي.

```
sendData = sentence.getBytes();
```

يقوم هذا السطر أساساً بتحويل نوع البيانات، حيث يأخذ كائناً نصياً (سلسلة حروف) يسمى sentence ويحوله إلى مصفوفة بايتات تخزّن في sendData.

```
DatagramPacket sendPacket = new DatagramPacket(sendData,
sendData.length, IPAddress, 9876);
```

يبني هذا السطر الرزمة sendPacket، التي سيدفعها الزبون إلى شبكة الإنترنت عبر مقبسه. تتضمن تلك الرزمة البيانات الموجودة في sendData، وطول هذه البيانات، وعنوان IP للخادم، ورقم منفذ الخادم (والذي وضعناه 9876 في هذا المثال). لاحظ أن sendPacket كائن من النوع DatagramPacket.

```
clientSocket.send(sendPacket);
```

في هذا السطر تأخذ الوظيفة ()send للكائن clientSocket الرزمة (التي بُنيَت للتو) وترسلها إلى الشبكة عبر clientSocket. ومرة أخرى لاحظ أن UDP يُرسِل

سطراً من الحروف بأسلوب مختلف جداً عن أسلوب TCP. ببساطة قام TCP بإدخال سلسلة الحروف إلى "المجرى" الذي له اتصال منطقي مباشر بالخادم؛ أما UDP فإنه يُكوِّن رزمة تتضمن عنوان الخادم. بعد إرسال الرزمة ينتظر الزبون استلام رزمة من الخادم.

```
DatagramPacket receivePacket = new DatagramPacket(receiveData,
receiveData.length);
```

في هذا السطر (بينما ينتظر الزبون وصول رزمة من الخادم) يُنشئ الزبون مكاناً لحفظ الرزمة (placeholder) يسمى receivePacket من النوع DatagramPacket. ثم يدخل الزبون طور خمول إلى أن يتلقى رزمة؛ عند ذلك يضع الرزمة في receivePacket باستخدام الأمر التالي:

```
clientSocket.receive(receivePacket);
String modifiedSentence = new String(receivePacket.getData());
```

يستخلص هذا السطر البيانات من receivePacket ويجري تحويلاً للنوع (type conversion) من مصفوفة بايتات إلى كائن نصي يسمى modifiedSentence.

```
System.out.println("FROM SERVER:" + modifiedSentence);
```

يعرض هذا السطر (والموجود أيضاً في نسخة TCPClient) محتويات modifiedSentence على شاشة الزبون.

```
clientSocket.close();
```

يغلق هذا السطرالأخير المقبس. ولأن UDP غير توصيلي، لا يؤدي ذلك إلى إرسال الزبون رسالة من طبقة النقل إلى الخادم (على النقيض من TCPClient).

برنامج الخادم: UDPServer.java

دعنا الآن نلقي نظرة على جانب خادم التطبيق:

```
import java.io.*;
import java.net.*;
class UDPServer {
    public static void main(String argv[]) throws Exception {
        DatagramSocket serverSocket = new DatagramSocket(9876);
        byte[] receiveData = new byte[1024];
        byte[] sendData = new byte[1024];
        while(true) {
            DatagramPacket receivePacket = new DatagramPacket(
            receiveData, receiveData.length);
            serverSocket .receive(receivePacket);
            String sentence = new String(receivePacket.getData());
            InetAddress IPAddress = receivePacket.getAddress();
            int port = receivePacket.getPort();
            String capitalizedSentence = sentence.toUpperCase();
            sendData = capitalizedSentence.getBytes();
            DatagramPacket sendPacket = new DatagramPacket(sendData,
            sendData.length, IPAddress, port);
            serverSocket .send(sendPacket);
        }
    }
}
```

يُنشئ برنامج UDPServer.java مقبساً واحداً (كما هو موضح في الشكل 2-36). يسمى المقبس serverSocket، وهو كائن من نوع DatagramSocket تماماً كما كان المقبس في جانب الزبون. ومرةً أخرى لا توجد كائنات "مجرى" مرتبطة بالمقبس. دعنا الآن نلقي نظرة على سطور الكود التي تختلف عن TCPServer.java.

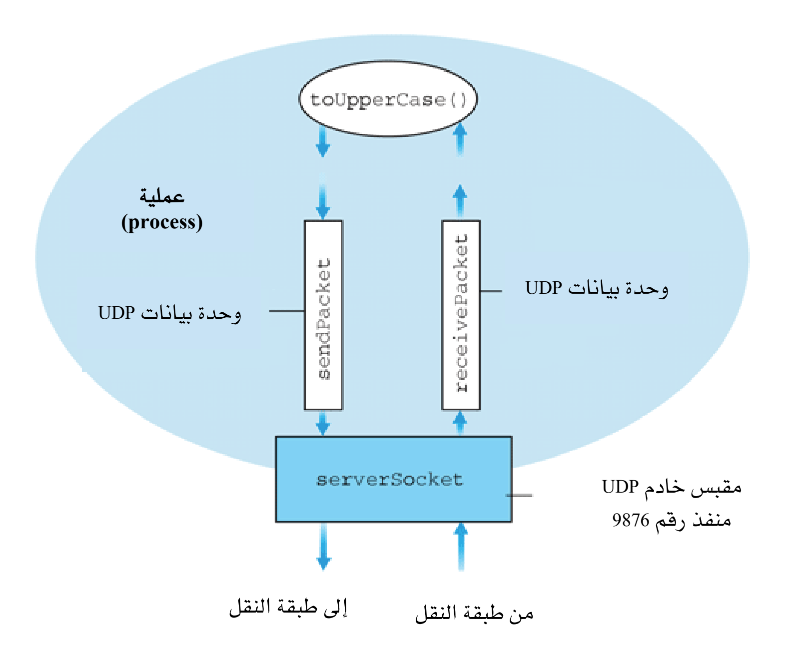
```
DatagramSocket serverSocket = new DatagramSocket(9876);
```

يُنشئ هذا السطر كائن serverSocket من النوع DatagramSocket على المنفذ رقم 9876. سوف تمر كل البيانات التي ترسل أو تستقبل خلال ذلك المقبس. ولأن UDP بروتوكول غير توصيلي، فليس من الضروري إنشاء مقبس جديد والاستمرار في الإنصات لطلبات اتصال جديدة (كما فعلنا في TCPServer.java). وإذا اتصل عدد من الزبائن بذلك التطبيق، فسيُرسل الجميع رزمهم عبر هذا الباب الوحيد serverSocket.

```
String sentence = new String(receivePacket.getData());
InetAddress IPAddress = receivePacket.getAddress();
int port = receivePacket.getPort();
```

تستخرج السطور الثلاثة أعلاه الرزمة التي تصل من الزبون. في السطر الأول، تُنتزع البيانات من الرزمة وتوضع في سلسلة الحروف sentence (وهذا السطر له مماثل في UDPClient). ينتزع السطر الثاني عنوان IP؛ وينتزع السطر الثالث رقم منفذ الزبون الذي يُختار اعتباطياً من قبل الزبون وهو يختلف عن رقم منفذ الخادم 9876 (سوف نُناقش أرقام منافذ الزبون بشيء من التفصيل في الفصلِ القادم). من الضروري للخادم أن يحصل على عنوان الزبون (عنوان IP ورقم المنفذ) لكي يمكنه الرد على رسالة الزبون.

بهذا ينتهي تحليلنا لزوج برامج UDP. ولكي تختبر التطبيق قم بإنشاء الملفات UDPClient.java على مضيف وUDPServer.java على مضيف آخر. (تأكّد من تضمين اسم المضيف الصحيح للخادم في برنامج الزبون UDPClient.java). قم بترجمة وتنفيذ البرنامجين على مضيفاتهما الخاصة. بخلاف TCP يمكنك تنفيذ جانب الزبون قبل تنفيذ جانب الخادم، وذلك لأن عملية الزبون لا تحاول بدء الاتصال بالخادم عند تنفيذ برنامج الزبون. ويمكنك استخدام التطبيق بكتابة جملة من برنامج الزبون، فيرسلها إلى الخادم، ثم يستلم الرد ويطبعه على الشاشة.



الشكل 2-36 ليس لدى الخادم UDPServer أي مجارٍ؛ يستلم المقبس رزماً من العملية ويسلمها رزماً.

2-9 الخلاصة

درسنا في هذا الفصل مفاهيم وسمات تطوير تطبيقات الشبكة. تعلمنا عن البنية المعمارية "زبون/خادم" والموجودة في كل مكان والمستعملة في العديد من تطبيقات الإنترنت، ورأينا استخداماتها في بروتوكولات HTTP، وFTP، وSMTP، وPOP3، وDNS. كما درسنا تلك البروتوكولات الهامة وتطبيقاتها المناظرة كالويب ونقل الملفات والبريد الإلكتروني وDNS بشيء من التفصيل. تناولنا أيضاً بنية النظائر التي يزداد انتشارها باضطراد وكيفية استخدامها في العديد من التطبيقات. كما استعرضنا كيف يمكن استخدام واجهة برمجة المقبس API لبناء تطبيقات الشبكة. كما تناولنا استخدام المقابس لتوفير خدمات

النقل من طرف إلى طرف سواء التوصيلي منها (TCP) أو اللاتوصيلي (UDP). إن الخطوة الأولى في رحلتنا خلال البنية المعمارية الطبقية للشبكة قد اكتملت الآن!

في بداياتنا الأولى مع هذا الكتاب (وبالتحديد في الجزء 1-1) أوردنا تعريفاً مبسطاً وغير واضح بعض الشيء للبروتوكول على أنه "صيغ الرسائل التي يتم تبادلها بين اثنين أو أكثر من الكيانات المتصلة، بالإضافة إلى الخطوات التي تتخذ عند إرسال أو استقبال رسالة أو حصول حدث آخر." لا شك أن المادة العلمية في هذا الفصل وبشكلٍ خاص دراستنا المُفصَّلة لبروتوكولات التطبيقات المختلفة قد أعطت الآن الكثير من المعاني لهذا التعريف، فالبروتوكولات مفهوم أساسي في مجال الشبكات، ودراستنا لبروتوكولات التطبيقات أعطتنا الفرصة الآن لتطوير فهم أكثر وعياً ودراية بماهية البروتوكولات وأهميتها.

تناولنا في الجزء 2-1 نماذج الخدمة التي يقدمها بروتوكولا طبقة النقل TCP وUDP للتطبيقات التي تستخدمهما، وألقينا نظرة عن قرب على تلك النماذج عندما طوَّرنا تطبيقات بسيطة تعمل على TCP وUDP في الجزء 2-7 والجزء 2-8 على الترتيب. ولكننا لم نسهب القول حول كيفية توفير TCP وUDP لتلك النماذج. على سبيل المثال عرفنا أن TCP يوفر خدمة نقل للبيانات موثوقة، ولكننا لم نذكر بعد كيفية تحقيق ذلك. في الفصل القادم سوف نُلقي نظرة أدق نستعرض من خلالها ليس فقط "ماذا" ولكن أيضاً "كيف" و"لماذا" فيما يتعلق ببروتوكولات النقل. بعد تلك المعرفة حول تركيب تطبيقات الإنترنت وبروتوكولات طبقة التطبيقات، يمكننا الآن أن نغوص أعمق في رصّة بروتوكولات الإنترنت وفحص طبقة النقل في الفصل القادم.

أسئلة وتمارين وتدريبات الفصل الثاني

❖ أسئلة مراجعة

- **الجزء 2-1**

1. اذكر أسماء خمسة تطبيقات ذات ملكية عامة للإنترنت واذكر أسماء بروتوكولات طبقة التطبيقات التي يستخدمونها.
2. ما الفرق بين بنية الشبكة المعمارية وبنية التطبيقات؟
3. في جلسة اتصال بين عمليتين، أي منهما يمثل الزبون وأي منهما يمثل الخادم؟
4. في تطبيق مشاركة النظائر للملفات، هل توافق على أنه "لا يوجد ما يمثل جانبي الزبون والخادم في جلسة الاتصال"؟ بيّن سبب الموافقة أو الرفض.
5. ما المعلومات التي تستخدمها عملية يتم تشغيلها على أحد المضيفات لتعريف عملية أخرى يجري تنفيذها على مضيف آخر؟
6. افترض أنك تريد أن تقوم بتنفيذ معاملة تجارية (transaction) معينة على خادم ما من مضيف بعيد بأقصى سرعة ممكنة، فهل ستستخدم بروتوكول TCP أو UDP؟ ولماذا؟
7. بالرجوع إلى الشكل 2-4 نرى أنه لا توجد تطبيقات مذكورة بالشكل تضع قيوداً على كلٍّ من التوقيت وفقد البيانات معاً، فهل تستطيع تخيل أحد تلك التطبيقات التي تتطلب عدم فقد للبيانات وتكون شديدة الحساسية للتوقيت؟
8. اذكر الأربع فئات العامة للخدمات التي يمكن أن يوفرها بروتوكول طبقة النقل؛ مع بيان اسم البروتوكول المستخدم مع كل فئة من فئات الخدمة هل هو TCP أو UDP أو كلاهما؟
9. تذكر أنه يمكن تحسين بروتوكول TCP باستخدام بروتوكول SSL لتوفير خدمات الأمن (بما في ذلك التشفير) بين العمليتين المتصلتين؛ فهل يعمل بروتوكول SSL في طبقة النقل أم في طبقة التطبيقات؟ وماذا يجب على مطوّر التطبيقات أن يفعل إذا أراد أن يستخدم SSL؟

- **الأجزاء 2-2 حتى 2-5**

10. ما المقصود ببروتوكول مصافحة؟

11. لماذا يستخدم بروتوكول TCP وليس UDP مع كلٍّ من البروتوكولات التالية: HTTP، FTP، SMTTP، POP3؟
12. افترض أن أحد مواقع التجارة الإلكترونية يريد الاحتفاظ بسجل المشتريات لكل زبون؛ صف كيف يمكن تنفيذ ذلك باستخدام الكوكيز.
13. صف كيف يمكن أن يؤدي استخدام ذاكرة الويب المخبأة إلى تخفيض تأخير استلام المتصفّح للكائن المطلوب. هل يؤدي استخدام ذاكرة الويب المخبأة إلى تخفيض التأخير لكل الكائنات المطلوبة أم لبعض هذه الكائنات فقط؟ ولماذا؟
14. استخدم Telnet للاتصال بخادم ويب وأرسل رسالة طلب متعددة السطور وتتضمن سطر الترويسة If-Modified-Since لتتسبب في ظهور 304 Not Modified في سطر الحالة في رسالة الرد.
15. لماذا يقال أن FTP يرسل يرسل معلومات التحكم "خارج النطاق" (out-of-band)؟
16. افترض أن أليس ترسل رسالة بريد إلكتروني من خلال حسابها على نظام للبريد الإلكتروني مبني على الويب (كـ Hotmail وgmail) إلى بوب والذي يستخدم POP3 للوصول لبريده. ناقش كيف تنتقل الرسالة من مضيف أليس إلى مضيف بوب؛ مع ذكر سلسلة بروتوكولات طبقة التطبيقات المستخدمة أثناء انتقال الرسالة بين المضيفين.
17. استعرض سطور الترويسة لإحدى رسائل البريد الإلكتروني التي تلقيتها حديثاً. ما عدد سطور الترويسة التي تتضمن Received؟ حلل كل سطر من سطور الترويسة في تلك الرسالة.
18. من منظور المستخدم ما الفرق بين نمط download-and-delete ونمط download-and-keep في بروتوكول POP3؟
19. هل من الممكن أن يستخدم خادم الويب لمؤسّسة وخادم البريد الإلكتروني لها نفس الاسم البديل للمضيف (مثلاً foo.com)؟ ما نوع سجل المورد (RR) لمضيف البريد الإلكتروني؟

- **الجزء 2-6**

20. في بروتوكول BitTorrent افترض أن أليس تزود بوب بالقطع على أساس 30 ثانية؛ فهل من الضروري أن يقوم بوب برد الجميل بتزويدها القطع إلى أليس بنفس تلك المدة؟ بيّن السبب؟

21. افترض أن أليس تلتحق كنظير جديد مع BitTorrent بدون معالجة لأيٍّ من القطع. ولذا لا يمكنها أن تصبح واحداً من النظائر الأربعة على قمة قائمة النظائر المُحمِّلة (uploaders) لدى أيٍّ من النظائر الأخرى. بين كيف يمكنها عندئذ أن تحصل على أول قطعة رغم عدم وجود أي شيء لتحميله؟
22. ما المقصود بالشبكة الإضافية (overlay network)؟ هل تتضمن موجّهات؟ ما الروابط (edges) فيها؟ وكيف يتم إنشاء وتعديل الشبكة الإضافية لفيضان الاستفسار؟
23. من أي وجه يكون نظام الرسائل الفورية بفهرس مركزي نظاماً هجيناً من بنية زبون/خادم وبنية النظائر؟
24. تستخدم معظم أنظمة الرسائل الفورية اليوم فهرساً مركزياً لتحديد أماكن المُستخدِمين. افترض أنه بدل من ذلك استخدمت شبكة إضافية بفيضان الاستفسار (مثل Gnutella) لتحديد أماكن المُستخدِمين. صف كيف يمكن أن يتم ذلك، وناقش مميزات وعيوب مثل هذا التصميم.
25. يستخدم بروتوكول Skype طرق النظائر لوظيفتين هامتين؛ ما هما؟
26. اذكر على الأقل أربعة تطبيقات ذات طبيعة تناسبها بينة النظائر (أمثلة على ذلك توزيع الملفات والرسائل الفورية).

- **الأجزاء 2-7 حتى 2-8**

27. يحتاج خادم UDP الموصوف في جزء 2-8 لمقبس واحد فقط بينما يحتاج خادم TCP الموصوف في الجزء 2-7 لمقبسين؛ لماذا؟ وإذا كان على خادم TCP أن يدعم *n* من التوصيلات في نفس الوقت (كلٌّ منها من مضيف مختلف)، فكم عدد المقابس التي سيحتاجها خادم TCP؟
28. في تطبيق زبون/خادم على بروتوكول TCP الموصوف في الجزء 2-7، لماذا يجب أن يتم تشغيل الخادم قبل الزبون؟ ولماذا لا نحتاج لنفس الشرط في حالة تطبيق زبون/خادم على بروتوكول UDP الموصوف في الجزء 2-8؟

❖ تدريبات

1. صح أم خطأ

a. يطلب مستخدم صفحة ويب تتضمن بعض النصوص وصورتين. لهذه الصفحة سيرسل الزبون رسالة طلب واحدة ويستقبل ثلاث رسائل رد.

b. يمكن أن تُرسَل صفحتا ويب مختلفتان (مثلاً: www.mit.edu/research.html وwww.mit.edu/students.html) على نفس التوصيلة الدائمة.

c. من الممكن لقطعة TCP أن تحمل رسالتي طلب HTTP مختلفتين عند وجود توصيلات غير دائمة بين المتصفّح وخادم الأصل.

d. يدل سطرالترويسة Date في رسالة رد HTTP على زمن آخر تعديل للكائن المُتضمَّن في تلك الرسالة.

2. اقرأ RFC 959 عن بروتوكول FTP ثم اذكر كل أوامر الزبون التي يدعمها.

3. اعتبر زبون HTTP يريد أن يسترجع مستند ويب من عنوان URL معين. وبافتراض أن عنوان IP لخادم HTTP غير معروف في البداية. ما هي البروتوكولات المطلوبة في كلٍّ من طبقة النقل وطبقة التطبيقات بالإضافة إلى HTTP لتنفيذ هذه المهمة؟

4. اعتبر سلسلة حروف الأسكي (ASCII) التالية والتي تم إلتقاطها ببرنامج Ethereal عندما أرسل المتصفّح رسالة GET (أي أن هذا النص هو المحتوى الفعلي لرسالة GET). تمثل الأحرف <cr><lf> بداية السطر.

GET /cs453/index.html HTTP/1.1*<cr><lf>*Host: gai
a.cs.umass.edu*<cr><lf>*User-Agent: Mozilla/5.0 (
Windows;U; Windows NT 5.1; en-US; rv:1.7.2) Gec
Ko/20040804 Netscape/7.2 (ax) *<cr><lf>*Accept: ex
t/xml,application/xml,application/xhtml+xml,text
/html;q=0.9,text/plain;q=0.8,image/png,*/*;q=0.5
*<cr><lf>*Accept-Language: en-us,en;q=0.5*<cr><lf>*Accept-
Encoding: zip,deflate*<cr><lf>*Accept-Charset: ISO
-8859-1,utf-8;q=0.7,*;q=0.7*<cr><lf>*Keep-Alive: 300*<cr>*
*<lf>*Connection: keep-alive*<cr><lf><cr><lf>*

أجب على الأسئلة التالية مع بيان مكان الإجابة في رسالة GET:

a. ما عنوان URL للمستند المطلوب من المتصفّح؟

b. ما هو رقم الإصدار (النسخة) لبروتوكول HTTP الذي يستخدمه المتصفّح؟

c. ما نوع التوصيلة التي يطلبها المتصفّح هل هي دائمة (persistent) أم غير دائمة (non-persistent) ؟

d. ما هو عنوان IP للمضيف الذي يجري تشغيل المتصفّح عليه؟

5. يبيّن النص التالي الرسالة المُرسَلة من الخادم رداً على رسالة GET في السؤال السابق.

HTTP/1.1 200 OK<*cr*><*lf*>Date: Tue, 07 Mar 2006
12:39:45GMT<*cr*><*lf*>Server: Apache/2.0.52 (Fedora)
<*cr*><*lf*>Last-Modified: Sat, 10 Dec 2005 18:27:46
GMT<*cr*><*lf*>ETag: "526c3-f22-a88a4c80"<*cr*><*lf*>Accept-
Range: bytes<*cr*><*lf*>Content-Length: 3874<*cr*><*lf*>
Keep-Alive: timeout=max=100<*cr*><*lf*>Connection:
keep-alive<*cr*><*lf*>Content-Type: text/html; charset=
ISO-8859-1<*cr*><*lf*><*cr*><*lf*><!doctype html public "-
//w3c//dtdd html 4.0 transitional//en"><*lf*><html><*lf*>
<head><*lf*> <met http-equiv="Content-Type"
content="text/html; charset=iso-8859-1"<*lf*><meta
name="GENERATOR" content= "Mozilla/4.79 [en] (Windows NT
5.0; U) Netscape]"><*lf*> <title>CMPSCI 453 / 591 /
NTU-ST550A Spring 2005 homepage</title><*lf*></head><*lf*>
<much more document text following here (not shown)>

أجب على الأسئلة التالية مع بيان مكان الإجابة في الرسالة:

a. هل تمكن الخادم من أن يجد المستند المطلوب بنجاح أم لا؟ ومتى تم إرسال الرد؟

b. متى تم آخر تعديل للمستند؟

c. ما حجم (عدد البايتات) المستند الذي تم تضمينه في رسالة الرد؟

d. ما أول خمسة بايتات في المستند الذي تم تضمينه في رسالة الرد؟ هل وافق الخادم أن تكون التوصيلة دائمة؟

6. احصل على المواصفات HTTP/1.1 (من طلب التعليقات RFC 2616)، وأجب على ما يلي:

a. وضح الإجراء المُستخدَم للتأشير (signaling) بين الزبون والخادم لبيان أن توصيلة دائمة يتم إغلاقها. هل يمكن أن يقوم الزبون أو الخادم أو كلاهما بإرسال إشارة لإغلاق توصيلة؟

b. ما هي خدمات التشفير التي يوفرها HTTP؟

7. افترض أنك ضغطت على رابط داخل المتصفّح للحصول على صفحة ويب، وكان عنوان IP المرتبط بعنوان URL لها غير موجود بالذاكرة المخبأة على المضيف المحلي، ولذا يلزم البحث في دليل DNS للحصول على عنوان IP. افتراض أنه تم زيارة n خادم DNS قبل الحصول على عنوان IP، وأن الزيارات المتتابعة استغرقت RTT_1، RTT_2، ....، RTT_n على التوالي. وافترض أيضاً أن الصفحة المطلوبة تتضمن كائناً واحداً يتألف من نصاً بسيطاً بصيغة HTML. افرض أن RTT_0 تمثل زمن الرحلة ذهاباً وإياباً بين المتصفح

والخادم. وافترض زمن نقل الكائن يساوي صفراً. ما مقدار الوقت المنقضي من لحظة ضغط المُستخدِم على الرابط حتى لحظة حصوله على الكائن؟

8. بالإشارة إلى السؤال السابق وبافترض أن ملف HTML يتضمن عناوين لثلاث كائنات صغيرة على نفسس الخادم، فما مقدار الوقت اللازم في الحالات التالية:
   a. توصيلات HTTP غير دائمة وتوصيلات TCP غير متوازية؟
   b. توصيلات HTTP غير دائمة وتوصيلات TCP متوازية؟
   c. توصيلة HTTP دائمة؟

9. بالإشارة إلى الشكل 2-12 والذي فيه شبكة مؤسّسة موصلة بالإنترنت. افترض أن الحجم المتوسط للكائن 900000 بت وأن معدل الطلب المتوسط من متصفّحات المؤسّسة للخادمات الأصل 15 طلب/ثانية. افترض أيضاً أن الوقت المستغرق من لحظة توجيه الموجّه على جانب الإنترنت لطلب HTTP إلى لحظة حصوله على الرد يعادل ثانيتين في المتوسط (انظر الجزء 2-2-5). اعتبر زمن الاستجابة المتوسط الكلي يُمثّل بمجموع زمن التأخير المتوسط للوصول (أي التأخير بين موجّه الإنترنت وموجّه المؤسّسة) وزمن تأخير الإنترنت المتوسط. استخدم المعادلة التالية لحساب زمن تأخير الوصول المتوسط:

$$t = \frac{\Delta}{1-\Delta\beta}$$

حيث Δ تمثل الزمن المتوسط اللازم لإرسال كائن على وصلة الوصول وβ تمثل معدل وصول الكائنات لوصلة الوصول.
   a. احسب زمن الاستجابة المتوسط الكلي.
   b. الآن افترض استخدام ذاكرة مخبأة في شبكة المؤسّسة المحلية وأن معدل إصابة الهدف (hit rate) 0.4، ثم احسب زمن الاستجابة المتوسط الكلي في هذه الحالة.

10. افترض وجود وصلة قصيرة طولها 10 أمتار ومعدل الإرسال عليها 150 بت/ثانية في كلا الإتجاهين. افترض أن رزم البيانات طولها 100000 بت وأن رزم التحكم (مثل إشعار الاستلام والمصافحة) طولها 200 بت. افترض وجود N من التوصيلات المتوازية خلال تلك الوصلة وأن الحيز الترددي للوصلة يقسم بالتساوي بين تلك التوصيلات. افترض الآن استخدام بروتوكول HTTP وأن حجم كل كائن يتم تنزيله 100 كيلوبت وأن الكائن الذي يتم تنزيله في البداية يتضمن 10 مراجع لكائنات أخرى على نفس الخادم. هل استخدام التوصيلات المتوازية له مغزى في هذه الحالة؟ والآن افترض استخدام HTTP الدائم، هل تتوقع تحسن ملحوظ عما سبق؟ وضح إجابتك وبيّن السبب.

11. اكتب برنامج TCP بسيطاً لخادم يقبل سطور مدخلات من الزبون ويقوم بعرض تلك السطور على الشاشة (وحدة الإخراج القياسية). ويمكنك القيام بذلك بتعديل برنامج TCPServer.java الذي سبق شرحه في الجزء 2-7-2. قم بترجمة البرنامج وتنفيذه. على أي جهاز يحتوي برنامج المتصفّح قم بإعداد خادم الوكيل (proxy) في المتصفّح ليشير إلى المضيف الذي يتم تشغيل برنامخ الخادم الذي أعددته وكذلك قم بضبط رقم المنفذ أيضاً. سيرسل المتصفّح الآن رسائل GET إلى هذا الخادم والذي سيقوم بعرض الرسائل على الشاشة. باستخدام هذا النظام حدّد ما إذا كان المتصفّح يستخدم GET الشرطية للكائنات المُخزَّنة بالذاكرة المخبأة محلياً أم لا.
12. ما الفرق بين MAIL FROM في SMTP وFROM في رسالة البريد نفسها؟
13. قم بقراءة طلب التعليقات 1939 لبروتوكول POP3 ووضح الغرض من أمر UIDL.
14. افترض أنك تستخدم POP3 للوصول إلي بريدك الإلكتروني.

a. افترض أنك أعددت زبون POP للعمل في نمط "نزّل واحذف"، ثم أكمل الإجراء التالي:

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah …
S: …….. blah
S: .
?
?
```

b. افترض أنك أعددت زبون POP للعمل في نمط "نزّل واحتفظ"، ثم أكمل الإجراء التالي:

```
C: list
S: 1 498
S: 2 912
S: .
C: retr 1
S: blah blah …
S: …….. blah
S: .
?
?
```

c. افترض أنك أعددت زبون POP للعمل في نمط "نزّل واحتفظ"، استخدم الجزء (b) من السؤال وافترض أنك استعدت الرسائل 1 و2، ثم خرجت من POP وبعد خمس دقائق قمت بالاتصال مرة أخرى بخادم POP لاستعادة الرسائل الجديدة التي أرسلت لك. اكتب الرسائل والتعليمات التي سيتم تبادلها بين الخادم والزبون لتحقيق هذه المهمة.

15. أجب عن الأسئلة التالية:

a. ما هي قاعدة بيانات whois؟

b. استخدم قواعد بيانات whois على الإنترنت لتحصل على أسماء خادمين DNS، واذكر قاعدة بيانات whois التي استخدمتها.

c. استخدم الأمر nslookup على جهازك المحلي لإرسال استفسارات DNS إلى ثلاثة خادمات DNS: أحدهم يمثل خادم DNS المحلي والاثنان الآخران يمثلان ما وجدته في الجزء (b). جرب الاستفسار عن سجلات من أنواع مختلفة: A، NS، MX. لخص ما تحصل عليه.

d. استخدم الأمر nslookup للبحث عن خادم ويب له عناوين IP متعددة. هل خادم الويب لمؤسستك (مدرسة أو شركة) له عناوين IP متعددة؟

e. استخدم قاعدة بيانات ARIN لتحديد مدى عناوين IP المُستخدَم في جامعتك.

f. صف كيف يستخدم المهاجم قواعد بيانات whois وأداة nslookup للقيام بعمليات استطلاعية لشبكة مؤسستك قبل أن يشن هجومه عليها.

g. ناقش لماذا يجب أن تكون قواعد بيانات whois متاحة علناً.

16. افترض أن ملفاً حجمه 10 جيجابت يتم توزيعه إلى N من النظائر. إذا كان معدل تحميل الملف من الخادم u_s يساوي 20 ميجابت/ثانية، ومعدل تنزيل الملف لكل نظير d_i يساوي 1 ميجابت/ ثانية ومعدل تحميل الملف لكل نظير يساوي u. قم بإعداد رسم بياني يوضح أدنى زمن توزيع لكلٍ من بنية زبون/خادم وبنية النظائرعند كل زوج من القيم لـ N وu عندما N = 10، 100، 1000 وu = 200 كيلوبت/ثانية، 600 كيلوبت/ثانية، 1 ميجابت/ثانية.

17. افترض أن ملفاً حجمه F بت يتم توزيعه إلى N من النظائر باستخدام بنية زبون/خادم. افترض نموذج حركة الموائع (fluid model) أي أن الخادم يمكنه الإرسال لعدة نظائر في نفس الوقت وبمعدلات مختلفة طالما أن المعدل الكلي لا يزيد عن us.

c. افترض أن $u_s / N \le d_{\min}$، حدد أسلوباً للتوزيع بحيث يكون زمن التوزيع NF/u_s.

d. افترض أن $u_s / N \ge d_{\min}$، حدد أسلوباً للتوزيع بحيث يكون زمن التوزيع $F/d_{\min}$.

e. استنتج أنه بشكلٍ عام يحسب أدنى زمن توزيع من $\max\{NF / u_s, F / d_{\min}\}$.

18. افترض أن ملفاً حجمه F بت يتم توزيعه إلى N من النظائر باستخدام بنية النظائر، وافتراض النموذج السائل (fluid model). وللتبسيط افترض أن $d_{\min}$ كبيرة جداً بحيث لا يمكن أن يمثل الحيز الترددي لتنزيل الملف أي عائق.

a. افترض $u_s \le (u_s + u_1 + ... + u_N)/N$ ؛ حدد أسلوباً للتوزيع بحيث يكون زمن التوزيع F/u_s.

b. افترض $u_s \ge (u_s + u_1 + ... + u_N)/N$ ؛ حدّد أسلوباً للتوزيع بحيث يكون زمن التوزيع $NF/(u_s + u_1 + ... + u_N)$.

c. استنتج أنه بشكلٍ عام يحسب أدنى زمن توزيع من $\max\{F/u_s, NF/(u_s + u_1 + ... + u_N)\}$.

19. افترض أن شبكة إضافية (overlay network) بها عدد N من النظائر النشطة وبين كل زوج منها توصيلة TCP فعالة. افترض أيضاً أن توصيلة TCP تمر خلال M موجّه. ما عدد العقد والأحرف المناظرة في تلك الشبكة؟

20. في مناقشتنا للشبكات الإضافية باستخدام فيضان الاستفسارات في الجزء 2-6 وصفنا ببعض التفصيل كيف يلتحق نظير جديد بتلك الشبكة. في هذا السؤال نريد استكشاف ما سيحدث عندما يغادر نظير تلك الشبكة. افترض أن كل نظير مشترك يحتفظ في أي لحظة بتوصيلات TCP لأربعة نظائر مختلفة على الأقل. افترض أن النظير X والذي له خمس توصيلات TCP مع النظائر الأخرى يريد أن يغادر.

a. في البداية اعتبر حالة المغادرة رشيقة (graceful leave) أي أن النظير X يغلق بشكلٍ واضح تطبيقه وبالتالي ينهي بشكلٍ واضح توصيلاته الخمسة. ماذا سيفعل كلٌّ من النظائر الخمسة التي كان يتصل بها النظير X؟

b. افترض الآن أن النظير X يقطع اتصاله بالإنترنت بشكلٍ مفاجئ دون انذار سابق لجيرانه الخمسة أنه سينهي توصيلات TCP معهم؛ ماذا يحدث في هذه الحالة؟

21. في هذا السؤال سنستكشف التوجيه على المسار العكسي (reverse-path routing) لرسائل الإصابة للاستفسار (query hit) في فيضان الاستفسار. افترض أن أليس أرسلت رسالة استفسار، وأن بوب تلقى رسالة الاستفسار (والتي قد تكون قد تم توجيهها خلال عدة نظائر بينية) ولديه ملف يطابق الاستفسار.

a. تذكر أنه عندما يمتلك نظير ملفاً مطابق للاستفسار فإنه سيرسل رسالة إصابة خلال المسار العكسي لرسالة الاستفسار المناظرة. كتصميم بديل يمكن أن يؤسس بوب توصيلة TCP مباشرة مع أليس ويرسل رسالة الإصابة للاستفسار خلال تلك التوصيلة. ما هي مميزات وعيوب هذا التصميم البديل؟

b. عندما يقوم النظير أليس بتوليد رسالة استفسار، يقوم بإدخال رقم تعريفي فريد في حقل MessageID (معرّف الرسالة). عندما يمتلك النظير بوب تطابق، يُولِّد رسالة إصابة مستخدماً نفس الرقم التعريفي في حقل MessageID. صف كيفية استخدام النظائر لحقل MessageID وجداول التوجيه المحلية لتحقيق توجيه المسار العكسي.

c. تصميم آخر بديل لا يستخدم أرقام تعريف: عندما تصل رسالة استفسار إلى نظيرٍ ما، يقوم النظير بتضمين عنوان IP له في الرسالة قبل أن يقوم بتوجيهها. صف كيفية استخدام النظائر لهذه الآلية لتحقيق توجيه المسار العكسي.

22. في هذا السؤال تصميم شبكة إضافية هرمية فيها نظائر عادية ونظائر ممتازة ونظائر فوق الممتاز (super-duper).

a. افترض أن كل نظير فوق الممتاز مسؤول تقريباً عن 200 نظير ممتاز وأن كل نظير ممتاز مسؤول تقريباً عن 200 نظير عادي. ما عدد النظائر فوق الممتاز المطلوب لشبكة بها 4 ملايين نظير؟

b. ما المعلومات التي يحتمل أن يخزنها كل نظير ممتاز؟ ما المعلومات التي يحتمل أن يخزنها كل نظير فوق الممتاز؟ وكيف يتم البحث في مثل هذا التصميم ثلاثي المستوى؟

23. في فيضان الاستفسار الذي نوقش في الجزء 2-6 افترض أن كل نظير متصل بعدد N من الجيران في الشبكة الإضافية. افترض أيضاً أن قيمة حقل عدد العقد (node-count) في البداية K. افترض أن أليس أرسلت استفساراً. احسب الحد الأعلى لعدد رسائل الاستفسارات التي سترسل خلال الشبكة الإضافية.

24. قم بإعداد وترجمة برامج جافا TCPClient وUDPClient على مضيفٍ ما وبرامج TCPServer وUDPServer على مضيفٍ آخر.

a. افترض أنك قمت بتشغيل TCPClient قبل تشغيل TCPServer؛ فماذا سيحدث؟ ولماذا؟

b. افترض أنك قمت بتشغيل UDPClient قبل تشغيل UDPServer؛ فماذا سيحدث؟ ولماذا؟

c. ماذا يحدث إذا استخدمت أرقام منافذ مختلفة لكلٍّ من الزبون والخادم؟

25. افترض أننا قمنا باستبدال السطر

```
DatagramSocket clientSocket = new DatagramSocket();
```

بالسطر

```
DatagramSocket clientSocket = new DatagramSocket(5432);
```

في برنامج UDPClient.java؛ فهل يلزم تغيير UDPServer.java؟ ما أرقام المنافذ لمقابس الزبون والخادم؟ وماذا كانت تلك الأرقام قبل القيام بهذا التغيير؟

❖ أسئلة للمناقشة

1. ما سبب انتشار تطبيقات النظائر لمشاركة الملفات في رأيك؟ هل لأنها توزع ملفات الموسيقى والفيديو مجاناً (رغم كونه قد يكون بشكلٍ مخالف أو غير رسمي)؟ أم لأن العدد الكبير من خادماتها يستجيب بكفاءة للطلب الهائل للميجابايتات من البيانات؟ أم كل هذه الأسباب مجتمعة؟
2. اقرأ البحث [Biddle 2003] عن شبكة Darknet ومستقبل توزيع المحتوى. هل توافق على كل أراء المؤلفين؟ بين أسباب ذلك.
3. غالباً ما تستخدم مواقع التجارة الإلكترونية ومواقع الويب الأخرى قواعد بيانات خلفية. كيف تتصل خادمات HTTP مع قواعد البيانات الخلفية تلك؟
4. كيف يمكنك إعداد برنامج المتصفّح لديك لاستعمال الذاكرة المخبأة المحلية؟ وما الاختيارات المتاحة لديك لتلك الذاكرة؟
5. هل يمكنك إعداد برنامج المتصفّح لديك ليفتح عدة توصيلات لموقع ويب على التوازي في نفس الوقت؟ ما هي ميزات وعيوب وجود عدد كبير من توصيلات TCP تلك في نفس الوقت؟
6. رأينا أن مقابس TCP في الإنترنت تتعامل مع البيانات المُرسَلة كتدفق من البايتات (byte stream) في حين تتعرف مقابس UDP على فواصل الرسائل. اذكر ميزة وعيب لواجهة برمجة التطبيقات التي تتعامل مع البيانات المُرسَلة كتدفق من البايتات وتلك التي تتعرف وتحتفظ بفواصل الرسائل.
7. ما هو خادم Apache للويب؟ وماذا يكلف؟ وما هي الوظائف التي يوفرها حالياً؟
8. افترض أن المنظمات التي تضع معايير الويب قررت أن تغير اصطلاحات التسمية لكي يسمى كل كائن ويشار إليه باسم فريد لا يعتمد على موقعه (وهو ما يُعرف بـ URN). ناقش بعض القضايا المتعلقة بذلك.
9. هل توجد اليوم أي شركات لتوزيع برامج البث التلفزيوني الحية على الإنترنت؟ وإذا كانت هناك أي من تلك الشركات فهل تستخدم بنية زبون/خادم أم بنية النظائر؟

10. هل الشركات التي توفر خدمة الفيديو عند الطلب (video-on-demand) على الإنترنت اليوم تستخدم بنية النظائر؟
11. كيف يوفر سكايب (Skype) خدمة اتصال من الحاسب للهاتف (PC-to-Phone) للعديد من البلدان المختلفة؟
12. ما هي بعض زبائن BitTorrent المنتشرة اليوم؟

❖ تدريبات على برمجة المقابس

1. خادم ويب متعدد التفريعات (Multi-threaded Web Server): في نهاية هذا التدريب ستكون قد طورت خادم ويب متعدد التفريعات في لغة جافا يكون قادراً على خدمة طلبات متعددة على التوازي. ستحقق الإصدار 1.0 لبروتوكول HTTP كما هو مُعرّف في RFC 1945. يُنشئ بروتوكول HTTP/1.0 توصيلة TCP منفصلة لكل زوج من الطلب والرد. تقوم تفريعة (thread) منفصلة بمعالجة كلٍّ من هذه التوصيلات. توجد أيضاً تفريعة رئيسة يتم فيها استماع الخادم للزبائن التي تريد تأسيس توصيلات معه. ولتبسيط مهمة البرمجة، سنطور البرنامج على مرحلتين. في المرحلة الأولى ستكتب خادم متعدد التفريعات يقوم ببساطة بإظهار محتويات طلبات HTTP التي يتلقاها. بعد أن تتأكد من أن هذا البرنامج يعمل بشكلٍ صحيح ستضيف الكود المطلوب لتوليد الرد المناسب. وأثناء تطويرك للبرنامج يمكنك اختبار هذا الخادم مستعيناً بمتصفّح الويب لديك (مثل متصفّح اكسبلورر من مايكروسوفت)؛ لكن عليك أن تتذكر أن هذا الخادم لا يعمل على المنفذ المعياري 80 ولذا عليك أن تحدد رقم المنفذ المستخدم ضمن عنوان URL الذي تكتبه للمتصفّح. على سبيل المثال إذا كان اسم المضيف الذي يجري تشغيل هذا الخادم عليه host.someschool.edu وكنت مستخدماً المنفذ رقم 6789 للخادم وتريد أن تستعرض الملف index.html، عليك أن تكتب العنوان كالآتي:

http://host.someschool.edu:6789/index.html

عندما يصادف هذا الخادم خطأً يجب أن يرد برسالة تبين هذا الخطأ بصيغة HTML المناسبة لاستعراضها ضمن نافذة المتصفّح. يمكنك أن تجد التفاصيل الكاملة لهذا التدريب وكذلك أجزاء من البرنامج في لغة جافا من خلال موقع الويب لهذا الكتاب http://www.awl.com/kurose-ross

2. زبون البريد (Mail Client): في هذا التدريب ستقوم بتطوير وكيل مُستخدِم للبريد الإلكتروني في لغة جافا بالخصائص التالية:

- يوفر واجهة رسومية للمُستخدِم (GUI) تضم حقولاً لخادم البريد المحلي وعنوان البريد للمُرسِل وعنوان البريد للمُستقبِل وعنوان الرسالة والرسالة نفسها.
- يؤسس توصيلة TCP بين الزبون وخادم البريد المحلي، ويرسل أوامر SMTP لهذا الخادم، ويستقبل ويعالج أوامر SMTP من هذا الخادم.

يجب أن يكون شكل واجهة المُستخدِم كالآتي:

From	
To	
Subject	
Message	

Send Clear Ouit

ستطور وكيل المُستخدِم لكي يرسل رسالة البريد الإلكتروني لمُستقبِل واحد على الأكثر في كل مرة. وعلاوة على ذلك سيفترض وكيل المُستخدِم أن الجزء الخاص بالنطاق لعنوان البريد للمُستقبِل هو نفسه الاسم القانوني (canonical name) لخادم البريد للمُستقبِل (أي أن وكيل المُستخدِم لن يقوم بالبحث في دليل أسماء النطاقات عن سجل خادم البريد ولذا يجب أن يُدخِل المُستخدِم الاسم الحقيقي لخادم البريد). يمكنك الاطلاع على التفاصيل الكاملة لهذا التطبيق وأجزاء من البرامج بلغة جافا في موقع كتابنا هذا على الويب http://www.awl.com/kurose-ross.

3. تطبيق UDP Pinger: في هذا التدريب ستطور زبوناً وخادماً للبنج ويستخدمان بروتوكول UDP. تشبه الوظائف المتوفرة مع هذه البرامج برنامج البنج المعياري المتاح ضمن أنظمة التشغيل الحديثة. يعمل برنامج البنج المعياري بإرسال رسائل صدى ICMP والتي يقوم الجهاز البعيد بإرجاعها للمُرسِل. عندئذ يحدد المُرسِل زمن رحلة الذهاب والإياب (RTT) بينه وبين ذلك الجهاز المراد اختبار الاتصال به. لا توفر لغة جافا أية وظائف لإرسال واستقبال رسائل ICMP وهذا هو السبب أنه يلزمك تطوير هذه البرامج في طبقة

التطبيقات باستخدام مقابس ورسائل بروتوكول UDP. توجد التفاصيل الكاملة لهذا التطبيق وأجزاء من البرامج بلغة جافا في موقع كتابنا هذا على الويب http://www.awl.com/kurose-ross.

4. خادم وكيل الويب: في هذا التدريب ستطور خادماً بسيطاً لوكيل الويب وفيه أيضاً يتم تخزين صفحات الويب في ذاكرة مخبأة. يستقبل هذا الخادم رسائل GET من متصفّح الويب، ويوجه تلك الرسائل لخادم الويب للوجهة، ثم يستقبل رد HTTP من هذا الخادم ويوجهه للمتصفّح. يعتبر هذا الخادم بسيط جداً لوكيل الويب، فهو يتعامل فقط مع رسائل GET. ومع ذلك يستطيع هذا الخادم أن يتعامل مع كل أنواع الكائنات بما في ذلك ملفات الصور وليس فقط صفحات HTML. توجد التفاصيل الكاملة لهذا التطبيق وأجزاء من البرامج بلغة جافا في موقع كتابنا هذا على الويب http://www.awl.com/kurose-ross.

❖ تدريبات معملية على استخدام برنامج Ethereal

1. بروتوكول HTTP: بعد أن عرفنا مبادئ تشغيل برنامج Ethereal لالتقاط الرزم في التدريب الأول في نهاية الفصل الأول، يمكننا الآن استخدامه لفحص البروتوكولات أثناء تشغيلها. في هذا التدريب سنفحص العديد من خصائص بروتوكول HTTP مثل: رسائل GET الأساسية والرد عليها، وصيغة رسائل HTTP، واسترجاع ملفات HTML كبيرة، واسترجاع ملفات HTML متضمنة عناوين URL، والتوصيلات الدائمة وغير الدائمة، والتحقق والأمن في بروتوكول HTTP. وكما هو الحال مع كل تدريبات Ethereal يوجد الوصف الكامل لهذا التدريب من خلال موقع الكتاب http://www.awl.com/kurose-ross.

2. دليل أسماء النطاقات (DNS): في هذا التدريب سنفحص عن كثب جانب الزبون لخدمة دليل أسماء النطاقات (والتي تقوم بالتحويل ما بين أسماء المضيفات وعناوين IP لها). تذكر من الجزء 2-5 أن دور الزبون في DNS يعتبر بسيطاً نسبياً (يقوم الزبون بإرسال استفسار إلى خادم DNS المحلي ويستقبل الرد منه). يمكن إخفاء الكثير من التفاصيل عن زبون DNS بينما تتفاعل خادمات أسماء النطاقات ذات التنظيم الهرمي مع بعضها بطريقة تتابعية أو تكرارية للحصول على عنوان IP المناظر لاسم المضيف المعني. ومع ذلك يعتبر البروتوكول من منظور زبون DNS بسيطاً نوعاً ما. سنرى طريقة تشغيل DNS في هذا التدريب والذي يوجد الوصف الكامل له من خلال موقع الكتاب http://www.awl.com/kurose-ross.

طبقة النقــــل

The Transport Layer

محتويات الفصل:

- مقدمة وخدمات طبقة النقل
- التجميع والتوزيع
- بروتوكول النقل اللاتوصيلي: UDP
- أساسيات النقل الموثوق للبيانات
- بروتوكول النقل التوصيلي: TCP
- مبادئ التحكم في الازدحام
- التحكم في الازدحام في بروتوكول TCP
- الخلاصة

نظراً لموقعها المتوسط بين طبقة التطبيقات وبقية طبقات الشبكة، تمثل طبقة النقل جزءاً أساسياً من البنية الطبقية للشبكة، حيث تلعب الدور الحاسم المتمثل في توفير خدمات الاتصال مباشرةً للتطبيقات التي يجري تشغيلها على المضيفات. تتلخص الفلسفة التعليمية التي سنتبعها في هذا الفصل في التناوب ما بين مناقشة مبادئ طبقة النقل ومناقشة طرق تطبيق تلك المبادئ في البروتوكولات الحالية، مع التركيز كالمعتاد على بروتوكولات الإنترنت، وبشكلٍ خاص بروتوكولي طبقة النقل: TCP وUDP.

سنبدأ بمناقشة العلاقة بين طبقة النقل وطبقة الشبكة، لتمهيد الطريق لفحص الوظيفة الهامة الأولى لطبقة النقل – ألا وهي تمديد خدمة طبقة الشبكة (للتوصيل بين نظامين طرفيين) إلى خدمة توصيل بين عمليتين في طبقة التطبيقات يجري تشغيلهما على النظامين الطرفيين. سنوضح هذه الوظيفة في تغطيتنا لبروتوكول وحدة بيانات المُستخدِم ((User Datagram Protocol (UDP)) وهو بروتوكول نقل غير توصيلي على الإنترنت.

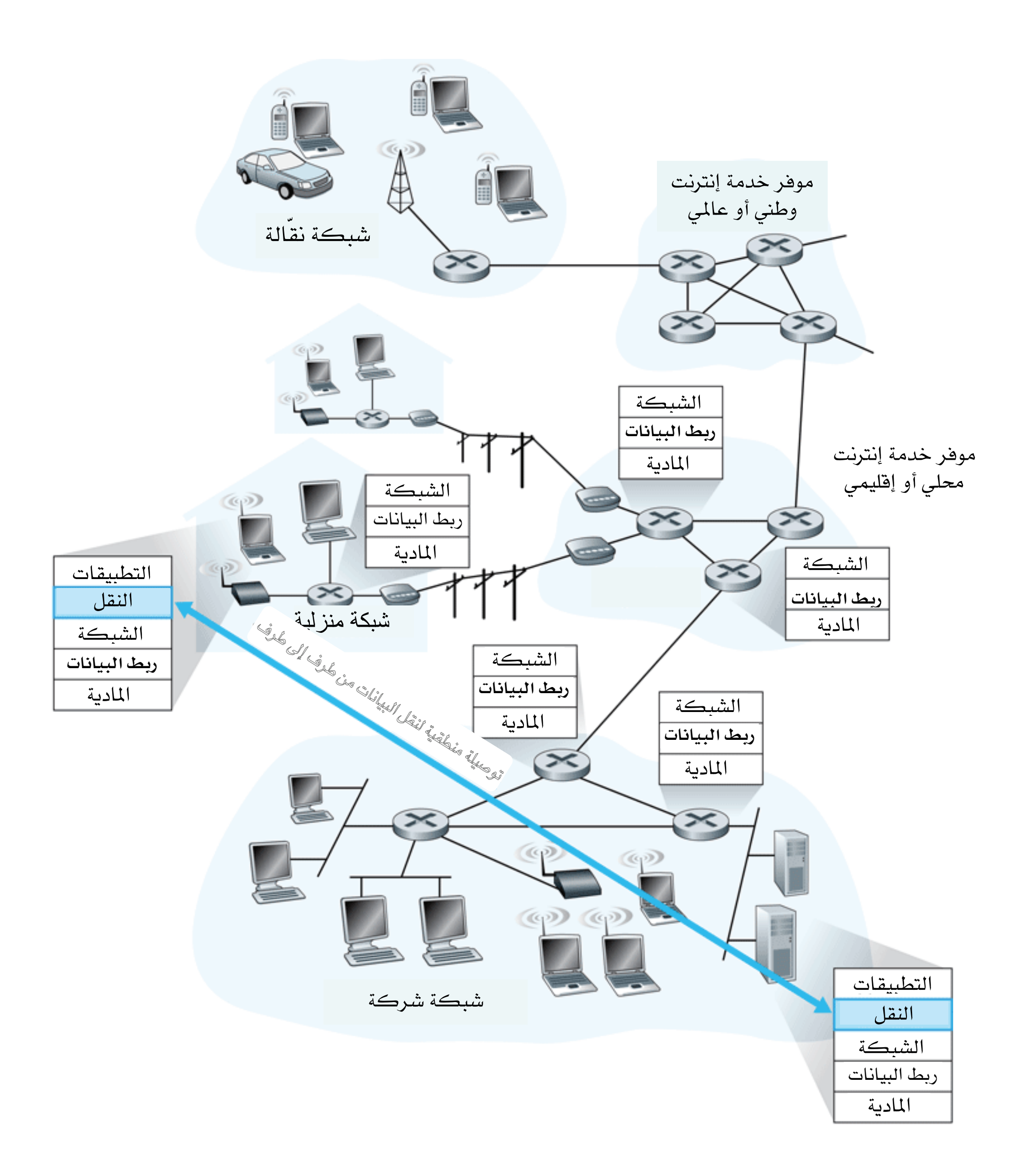
سنعود بعد ذلك إلى المبادئ ونواجه واحدةً من أهم المشاكل الأساسية في شبكات الحاسب: كيف يمكن لكيانين أن يتّصلا بشكلٍ موثوق عبر وسط قد يفقد أو يغيّر في البيانات؟ من خلال سلسلة سيناريوهات تزداد تعقيداً (وواقعية!) سنطور مجموعة من الأساليب التي تستخدمها بروتوكولات النقل لحل هذه المشكلة، وسنبين بعد ذلك كيف تتجسد هذه المبادئ في بروتوكول التحكم في الإرسال ((Transmission Control Protocol (TCP)) وهو بروتوكول نقل توصيلي على الإنترنت.

بعد ذلك سننتقل إلى المشكلة الأساسية الثانية في ربط الشبكات، ألا وهي التحكم في معدلات الإرسال في كيانات طبقة النقل لكي تتفادى الازدحام أو تتعافى من حدوثه في الشبكة، وسنتناول أسباب ونتائج الازدحام ونستعرض تقنيات التحكم في الازدحام المستعملة بكثرة. وبعد فهم الأمور المتعلقة بالتحكم في الازدحام فهماً جيداً، سندرس الطريقة التي يتبعها بروتوكول TCP لتحقيق ذلك.

1-3 مقدمة وخدمات طبقة النقل

في الفصلين السابقين أشرنا إلى دور طبقة النقل والخدمات التي توفرها. دعنا نراجع بسرعة ما تعلّمناه عن طبقة النقل. يوفر بروتوكول طبقة النقل اتصالاً منطقياً (logical communication) بين عمليات التطبيقات التي يجري تشغيلها على مضيفات مختلفة. نعني بالاتصال المنطقي أنه من وجهة نظر التطبيق تبدو المضيفات التي تُشغّل تلك العمليات كما لو كانت موصلةً مباشرة؛ في حين أنه في الواقع يمكن أن تقع المضيفات مادياً على جانبين متقابلين من المعمورة موصلةً عبر العديد من الموجّهات وبواسطة تشكيلة كبيرة من أنواع الوصلات. تستخدم عمليات طبقة التطبيقات الاتصال المنطقي الذي توفره طبقة النقل لإرسال الرسائل بين بعضها البعض، دون الاهتمام بتفاصيل البنية التحتية المادية المستخدمة لحمل تلك الرسائل. يوضح الشكل 3-1 مفهوم الاتصال المنطقي.

يتضح من هذا الشكل أن بروتوكولات طبقة النقل موجودة في الأنظمة الطرفية وليس في موجّهات الشبكة. تقوم طبقة النقل على الطرف المُرسِل بتحويل الرسائل التي تستلمها من عملية التطبيق إلى رزم طبقة النقل، والتي تُعرف في مصطلحات الإنترنت بقطع طبقة النقل (segments). يُحتمَل أن يتم ذلك بتجزئة رسائل التطبيق إلى أجزاء أصغر وإضافة ترويسة طبقة النقل إلى كل جزء منها لتكوين قطعة طبقة النقل. تقوم طبقة النقل بعد ذلك بدفع القطعة إلى طبقة الشبكة في نظام طرف الإرسال، حيث يتم تغليف القطعة للحصول على رزمة طبقة الشبكة (وحدة بيانات (datagram)) وإرسالها إلى وجهتها. من المهم ملاحظة أن موجّهات الشبكة تعمل فقط على حقول طبقة الشبكة في وحدة البيانات؛ أي أن تلك الموجّهات لا تفحص حقول قطعة طبقة النقل المغلّفة ضمن وحدة البيانات. وفي جانب الاستلام تنتزع طبقة الشبكة قطعة طبقة النقل من وحدة بيانات طبقة الشبكة وتدفع بالقطعة لأعلى إلى طبقة النقل. تقوم طبقة النقل بعد ذلك بمعالجة القطعة المُستلَمة، وتجعل البيانات في تلك القطعة متاحةً لاستخدام التطبيق في جهة الاستلام.



الشكل 3-1 توفر طبقة النقل توصيلة منطقية وليست مادية لنقل البيانات بين عمليات التطبيقات على نظامين طرفيين.

قد يتوفر أكثر من بروتوكول لطبقة النقل للاستخدام من قِبَل تطبيقات الشبكة. على سبيل المثال للإنترنت بروتوكولان: TCP وUDP. يوفر كلٌّ من هذين البروتوكولين مجموعةً مختلفةً من خدمات طبقة النقل للتطبيق الذي يستخدمه.

3-1-1 العلاقة بين طبقة النقل وطبقة الشبكة

تذكّر أن طبقة النقل توجد مباشرة فوق طبقة الشبكة في رصة البروتوكولات. وبينما يوفر بروتوكول طبقة النقل اتصالاً منطقياً بين عمليات (processes) تجري على مضيفات مختلفة، فإن بروتوكول طبقة الشبكة يوفر اتصالاً منطقياً بين المضيفات (hosts)، وهذا الفرق دقيق ولكنه مهم. دعنا نشرح هذا الفرق بالاستعانة بمثال.

تصور أن هناك بيتين، واحد على الساحل الشرقي والآخر على الساحل الغربي للولايات المتحدة، وكل بيت يأوي 12 طفلاً، والأطفال الذين في الساحل الشرقي أبناء عم الأطفال الذين في الساحل الغربي، وكلٌّ منهم يحب الكتابة إلى كل أبناء عمومته – حيث يكتب كل طفل إلى كل ابن أو بنت منهم خطاباً كل أسبوع، وكل خطاب يُرسَل في ظرف مستقل عن طريق خدمة البريد العمومية. وبهذا يُرسِل كل بيت 144 رسالة إلى البيت الآخر كل أسبوع، (بوسع هؤلاء الأطفال توفير الكثير من المال إذا استخدموا البريد الإلكتروني!). في كلٍّ من البيتين يتولى طفل واحد مسؤولية جمع البريد وتوزيعه: آن (Ann) في بيت الساحل الغربي وبيل (Bill) في بيت الساحل الشرقي. في كل أسبوع تقوم آن بزيارة كل إخوتها وأخواتها لجمع البريد ثم تسلمه إلى ساعي البريد التابع لخدمة البريد العمومية، والذي يقوم بزيارة البيت يومياً. عندما تصل رسائل إلى بيت الساحل الغربي، تتحمل آن أيضاً مسؤولية توزيع البريد الواصل على إخوتها وأخواتها. وفي المقابل يضطلع بيل بمهمات مماثلة على الساحل الشرقي.

في هذا المثال تزوّد خدمة البريد اتصالاً منطقياً بين البيتين – حيث تنقل تلك الخدمة الرسائل من بيت إلى بيت آخر، وليس من شخص إلى شخص آخر. من ناحية أخرى توفر آن وبيل اتصالاً منطقياً بين أبناء العم – حيث يقومان باستلام البريد من

وإلى إخوتهما وأخواتهما وتوزيعه عليهم. لاحظ أنه من منظور أبناء العم يُعتبر آن وبيل بمثابة خدمة البريد، رغم أنهما في الواقع مجرد جزء فقط (الجزء الموجود على النظام الطرفي) من عملية نقل البريد من طرف إلى طرف. هذا المثال التوضيحي يناظر العلاقة بين طبقة النقل وطبقة الشبكة:

- رسائل التطبيقات = الخطابات في الظروف
- العمليات = أبناء العمومة في البيتين
- المضيفات (الأنظمة الطرفية) = البيتان
- بروتوكول طبقة النقل = آن وبيل
- بروتوكول طبقة الشبكة = الخدمة البريدية (بما في ذلك سُعاة البريد)

استمراراً مع هذا التناظر، لاحظ أن آن وبيل يقومان بكل عملهما كلٌّ في حدود بيته فقط؛ فليس لهما أي صلةٍ مثلاً بتصنيف البريد في أيٍّ من مراكز البريد أو بنقل البريد من مركز بريد إلى آخر. بنفس الطريقة فإن بروتوكولات طبقة النقل تكمن في الأنظمة الطرفية. في النظام الطرفي يقوم نظام بروتوكول النقل بنقل الرسائل من عمليات التطبيقات إلى حافة الشبكة (أي طبقة الشبكة) والعكس بالعكس، لكنه ليس له أي رأي في كيفية نقل الرسائل ضمن قلب الشبكة. في الحقيقة كما يوضح الشكل 3-1، لا تتصرّف الموجّهات الوسيطة بناءً على أي معلومات قد تكون طبقة النقل قد أضافتها إلى رسائل التطبيقات، كما أنها لا تتعرف على تلك المعلومات.

واستمراراً مع القصة السابقة، لنفترض الآن أنه عندما تسافر آن وبيل في إجازة، فإن اثنين آخرين من أبناء العم (مثلاً سوزان وهاري) يحلان محلهما في توفير الخدمة الداخلية لجمع وتوزيع البريد للمجموعة في كل عائلة، لكن لسوء حظ العائلتين، قد لا يقومان بجمع وتوزيع البريد بالضبط بنفس طريقة آن وبيل لكونهما أصغر عمراً. فقد تقوم سوزان وهاري بجمع وتوزيع البريد مراتٍ أقل، كما أنهما قد يفقدان الرسائل من حين لآخر. وهكذا، فإن ابني العم سوزان وهاري لا يوفران نفس مجموعة الخدمات (أي نفس نموذج الخدمة) كآن وبيل. وبالمثل فإن شبكة

الحاسب قد يتوافر لديها عدة بروتوكولات للنقل، كلٌّ منها يوفر نموذج خدمة مختلف للتطبيقات التي تستخدمه.

واضح أن الخدمات التي يمكن لآن وبيل توفيرها محدودة بتلك التي يمكن للخدمة البريدية توفيرها. على سبيل المثال إذا كانت الخدمة البريدية لا تضمن حداً أقصى للمدة التي يستغرقها نقل البريد بين البيتين (ثلاثة أيام مثلاً) فلن يكون بوسع آن وبيل ضمان حدٍ أقصى للتأخير في توصيل البريد بين أيٍ من أبناء العمومة. وبالمثل فإن الخدمات التي يوفرها بروتوكول طبقة النقل غالباً ما تكون محدودة بنموذج الخدمة الذي يوفره بروتوكول طبقة الشبكة التحتي. إذا كان بروتوكول طبقة الشبكة لا يستطيع توفير ضمانات فيما يتعلق بالتأخير (delay) أو الحيّز الترددي (bandwidth) (ومن ثم معدّل أو سعة الإرسال) المتاح لقطع البيانات الخاصة بطبقة التطبيقات أثناء انتقالها بين المضيفات، فإن نظام طبقة النقل لن يستطيع توفير ضمانات عن التأخير أو الحيّز الترددي المتاح لرسائل التطبيقات التي يجري تبادلها بين العمليات.

ومع ذلك يمكن توفير بعض الخدمات من قِبَل بروتوكول النقل حتى عندما يكون بروتوكول الشبكة التحتي لا يوفر الخدمة المناظرة في طبقة الشبكة. على سبيل المثال، وكما سنرى في هذا الفصل، يمكن لبروتوكول النقل توفير خدمة موثوقة لنقل البيانات لتطبيقٍ ما حتى لو كان بروتوكول الشبكة التحتي غير موثوق، أي حتى لو كان بروتوكول الشبكة يفقد أو يُغير أو يكرر الرزم. كمثالٍ آخر (والذي سندرسه بتفصيل أكثر في الفصل الثامن أثناء تناولنا لأمن الشبكة)، يمكن لبروتوكول النقل استخدام التشفير لضمان عدم اطلاع الدخلاء على رسائل التطبيقات، حتى لو كانت طبقة الشبكة لا تستطيع ضمان سريّة قطع البيانات الخاصة بطبقة النقل.

3-1-2 فكرة عامة عن طبقة النقل في الإنترنت

تذكّر أن الإنترنت – وبشكلٍ عام شبكات TCP/IP – توفر بروتوكولين مختلفين في طبقة النقل لاستخدامات طبقة التطبيقات: أحدهما هو UDP

(بروتوكول وحدة بيانات المستخدم) والذي يوفر للتطبيق الذي يستخدمه خدمةً لاتوصيلية غير موثوقة، والآخر هو TCP (بروتوكول التحكم في الإرسال) والذي يوفر للتطبيق الذي يستخدمه خدمة نقل موثوق تعتمد على توصيلة. عند تصميم تطبيق للاستخدام على شبكة، يتعين على مطور التطبيق تحديد أيٍّ من هذين البروتوكولين سيستخدمه في طبقة النقل. وكما رأينا في الأجزاء 2-7 و2-8 يختار مطور التطبيقات ما بين UDP وTCP عند إنشاء المقابس.

لتبسيط المصطلحات ولتقليل التشويش والخلط (في كتاب تمهيدي كهذا عن شبكات الحاسب) سنشير إلى رزمة بيانات طبقة النقل كقطعة (segment) سواءً كان بروتوكول طبقة النقل TCP أو UDP. ومع ذلك فجدير بالذكر أن أدبيات الإنترنت (على سبيل المثال طلبات التعليقات RFCs) تسمّي رزمة طبقة النقل قطعة (segment) في حالة بروتوكول TCP فقط، بينما تستخدم التعبير "وحدة البيانات" (datagram) في حالة بروتوكول UDP. غير أن أدبيات الإنترنت نفسها تستخدم "وحدة البيانات" للتعبير أيضاً عن رزمة طبقة الشبكة! لكننا سنقصر استخدام التعبير "وحدة البيانات" (datagram) على رزمة بيانات طبقة الشبكة فقط.

قبل المضي قدماً في مقدمتنا القصيرة عن البروتوكولين TCP وUDP، من المفيد أن نذكر بضع كلمات عن طبقة الشبكة بالإنترنت (والتي سندرسها بالتفصيل في الفصل الرابع). يطلق على بروتوكول طبقة الشبكة بالإنترنت اسم بروتوكول الإنترنت (IP). يوفر هذا البروتوكول اتصالاً منطقياً بين المضيفات. نموذج الخدمة لبروتوكول IP هو نموذج "أفضل جهد" للتوصيل. بمعنى أن البروتوكول سيبذل "جهده الأقصى" لتوصيل قطع البيانات بين المضيفات المتصلة، ولكنه لا يقدم أي ضمانات بهذا الصدد. وبشكلٍ خاص فإنه لا يضمن وصول قطع البيانات، ولا يضمن توصيل القطع بالترتيب، ولا يضمن سلامة البيانات في تلك القطع من الأخطاء. لهذه الأسباب يقال: إن بروتوكول IP يوفر خدمة غير موثوقة (unreliable). من المفيد أيضاً أن نذكر هنا أن لكل مضيف عنواناً واحداً على الأقل من عناوين طبقة الشبكة، وهو ما يعرف بعنوان IP (سندرس عناوين IP بالتفصيل في الفصل الرابع).

بعد هذه اللمحة عن نموذج خدمة بروتوكول IP، دعنا نلخص الآن نماذج الخدمة التي يوفرها البروتوكولان UDP وTCP. إن المسؤولية الأساسية الأولى لهذين البروتوكولين هي تمديد خدمة التوصيل التي يوفرها بروتوكول IP بين نظامين طرفيين إلى خدمة توصيل بين عمليتين يجري تشغيلهما على النظامين الطرفيين. يُطلق على تمديد خدمة التوصيل من مضيف إلى مضيف إلى خدمة توصيل من عملية إلى عملية: التجميع (multiplexing) والتوزيع (demultiplexing) في طبقة النقل. سنتناول التجميع والتوزيع في طبقة النقل في الجزء التالي من هذا الفصل. يقوم كلٌّ من البروتوكولين UDP وTCP بتدقيق سلامة البيانات أيضاً عن طريق حقول خاصة باكتشاف الخطأ في الترويسة التي تُلحَق بقطعة البيانات.

تُعدّ خدمتا الحد الأدنى هاتان (نقل البيانات بين العمليات واكتشاف الخطأ) الخدمات الوحيدة التي يوفرها بروتوكول UDP. وبالتحديد يلاحظ أن بروتوكول UDP – مثله في ذلك مثل بروتوكول IP – يوفر خدمة غير موثوقة؛ فهو لا يضمن أن البيانات المُرسَلة من قِبَل عمليةٍ ما ستصل سليمة (أو ستصل على الإطلاق!) إلى العملية المقصودة على مضيف الوجهة. سنتناول بروتوكول UDP بالتفصيل في الجزء 3-3.

على الناحية الأخرى يوفر بروتوكول TCP عدة خدمات إضافية للتطبيقات. فهو يوفر – أولاً وقبل كل شيء – نقلاً موثوقاً للبيانات من خلال استعمال أساليب لضبط تدفق البيانات، والأرقام المتسلسلة للرزم، وأرقام إشعارات الاستلام، والموقّتات (وهي أساليب سنتناولها بالتفصيل في هذا الفصل). كما أنه يضمن توصيل البيانات المُرسَلة من عملية ما إلى العملية المقصودة على مضيف الوجهة صحيحة وغير مكرّرة وبنفس الترتيب. وهكذا يحوّل بروتوكول TCP خدمة IP غير الموثوقة بين الأنظمة الطرفية إلى خدمة موثوقة لنقل البيانات بين العمليات. كما يوفر بروتوكول TCP أيضاً تحكماً في الازدحام، وهذا الإجراء لا يعتبر خدمةً للتطبيق الذي يستخدم البروتوكول فقط بقدر ما هو خدمة للإنترنت ككل (أي خدمة للصالح العام). فبشكلٍ عام يمنع تحكم TCP في الازدحام أي توصيلة TCP من إغراق الوصلات والموجّهات بين المضيفات المتصلة بكمية مفرطة من حركة

مرور البيانات. ويسعى بروتوكول TCP لإعطاء كل توصيلة TCP تعبر وصلة مزدحمة نصيباً متساوياً من الحيّز الترددي الكلي المتاح للوصلة. يتم ذلك بتنظيم المعدلات التي تستخدمها جهات الإرسال التابعة لتوصيلات TCP في إرسال البيانات إلى الشبكة. ومن ناحية أخرى يُلاحَظ أن بروتوكول UDP لا ينظم حركة مرور بياناته، فالتطبيق الذي يستخدمه بوسعه استخدام أي معدل إرسال يرغبه وللمدة التي يرغبها.

أي بروتوكول يوفر نقلاً موثوقاً للبيانات وسيطرةً على الازدحام سيكون معقداً بالضرورة. سنحتاج إلى عدة أجزاء لتغطية مبادئ نقل البيانات الموثوق والتحكم في الازدحام، وأجزاء إضافية لتغطية بروتوكول TCP نفسه. سنتناول هذه المواضيع في الأجزاء من 3-4 إلى 3-8. وتتلخص طريقتنا في عرض هذه الموضوعات في هذا الفصل في التناوب مابين المبادئ الأساسية وتفاصيل البروتوكول نفسه. على سبيل المثال، سنناقش أولاً النقل الموثوق للبيانات بشكلٍ عام، ثم ننتقل لنوضح كيف يحقق بروتوكول TCP على وجه التحديد نقلاً موثوقاً للبيانات. وبنفس الطريقة سنناقش التحكم في الازدحام بشكلٍ عام أولاً ثم بعد ذلك نناقش كيف يتم التحكم في الازدحام في بروتوكول TCP. ولكن قبل التعرض لكل تلك المادة الجيدة، دعنا أولاً نلقي نظرة على خدمتي التجميع والتوزيع في طبقة النقل.

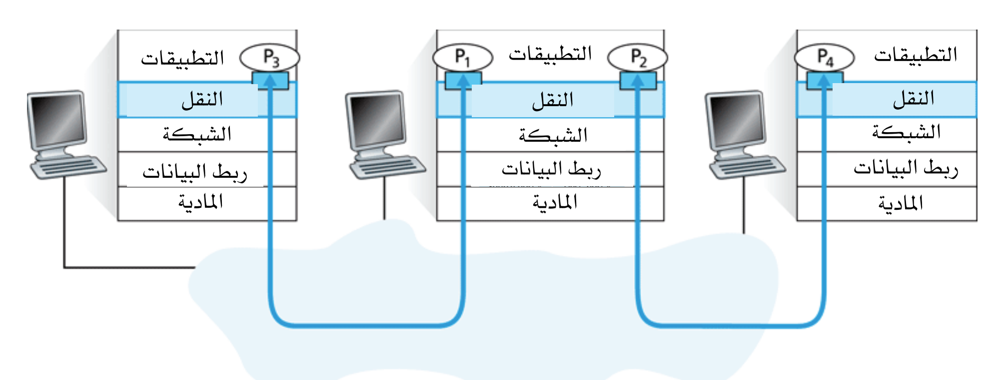
3-2 التجميع والتوزيع

نتناول في هذا الجزء خدمتي التجميع والتوزيع في طبقة النقل، أي تمديد خدمة التوصيل من مضيف إلى مضيف التي توفرها طبقة الشبكة إلى خدمة توصيل من عملية إلى عملية والتي توفرها طبقة النقل للتطبيقات التي يجري تشغيلها على المضيفات. وللحفاظ على التحديد في تناول الموضوع، سنناقش هذه الخدمة الأساسية لطبقة النقل ضمن سياق الإنترنت، إلا أننا نؤكد على أن التجميع والتوزيع خدمتان مطلوبتان لكل شبكات الحاسب.

في مضيف الوجهة: تتسلم طبقة النقل قطع البيانات من طبقة الشبكة التي تقع تحتها مباشرةً، وتتحمل مسؤولية توصيل البيانات الموجودة في تلك القطع إلى

عمليات التطبيقات الملائمة التي يجري تشغيلها في المضيف. لنلقِ نظرة على مثال. افترض أنك جالس أمام حاسبك تتجول بين صفحات الويب بينما تقوم في نفس الوقت بتشغيل عملية أخرى لتنزيل ملفات عبر بروتوكول FTP وتشغيل عمليتين أُخرَيَيْن للدخول على حاسب عن بعد عبر بروتوكول Telnet. لديك إذن أربع عمليات شغالة من تطبيقات الشبكة – عمليتا Telnet، وعملية FTP، وعملية HTTP. عندما تتسلم طبقة النقل في حاسبك البيانات من طبقة الشبكة تحتها، فإنها تحتاج لتوجيه تلك البيانات إلى إحدى تلك العمليات الأربع. دعنا الآن نرى كيف يتم ذلك.

لعلك تذكر من الأجزاء 2-7 و2-8 أن العملية (كجزء من تطبيق الشبكة) يمكن أن يكون لها مقبسٌ أو أكثر، أي بوابات تمر عبرها البيانات من الشبكة إلى العملية ومن العملية إلى الشبكة. وبالتالي كما يوضح الشكل 3-2 فإن طبقة النقل في مضيف الاستقبال لا تسلّم البيانات في الحقيقة مباشرة إلى العملية، ولكن بدلاً من ذلك تسلمها إلى مقبس متوسط. ونظراً لأنه في أي وقت يمكن أن يكون هناك أكثر من مقبس واحد في مضيف الاستقبال، فإن كل مقبس له مُعرّف (identifier) يميزه. تعتمد صيغة المُعرّف على ما إذا كان المقبس مقبس UDP أو TCP، كما سنرى بعد قليل.



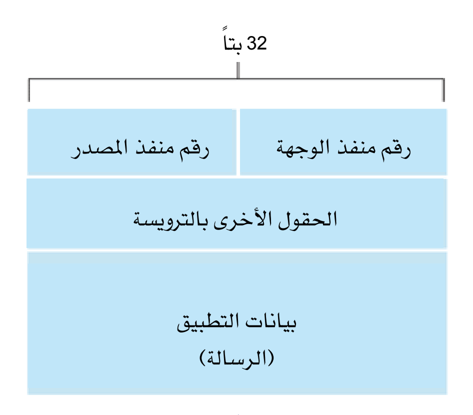
الشكل 3-2 عمليتا التجميع والتوزيع في طبقة النقل.

لنتأمل الآن كيف يقوم مضيف استقبال بتوجيه قطعة بيانات وصلت لطبقة النقل إلى المقبس الملائم. تتضمن قطعة بيانات طبقة النقل عدة حقول لهذا الغرض. في طرف الاستقبال تفحص طبقة النقل تلك الحقول لتحدد مقبس الاستلام ومن ثم توجّه القطعة إلى ذلك المقبس. يطلق على وظيفة توصيل البيانات الموجودة في قطعة طبقة النقل إلى المقبس الصحيح اسم "التوزيع" (demultiplexing). وبالمثل يطلق اسم "التجميع" (multiplexing) على وظيفة تجميع أجزاء البيانات في مضيف المصدر من المقابس المختلفة وتغليف كل جزء من البيانات بمعلومات الترويسة (التي ستستعمل لاحقاً في عملية التوزيع) لتكوين قطع بيانات طبقة النقل ثم دفع القطع إلى طبقة الشبكة. لاحظ أن طبقة النقل في المضيف المتوسّط في الشكل 3-2 يجب أن توزع قطع البيانات الواصلة من طبقة الشبكة تحتها لأي من العمليتين P1 أو P2 فوقها. يتم ذلك بتوجيه قطع البيانات الواصلة إلى مقبس العملية المناظرة. يجب على طبقة النقل في المضيف المتوسّط أيضاً تجميع البيانات الخارجة من تلك المقابس، وتكوين قطع بيانات طبقة النقل، ودفع تلك القطع لأسفل إلى طبقة الشبكة. ورغم أننا قدّمنا خدمتي التجميع والتوزيع في سياق بروتوكول النقل على الإنترنت، فإنه من المهم إدراك أنهما مطلوبتان طالما كان هناك بروتوكول واحد في طبقةٍ ما (طبقة النقل أو غيرها) يتم استخدامه من قِبَل عدة بروتوكولات في الطبقة الأعلى مباشرةً.

لتوضيح وظيفة التوزيع تذكّر مثال البيتين في الجزء السابق. يتم تمييز كل طفل أو طفلة من الأطفال باسمه أو اسمها. عندما يستلم بيل رزمة بريد من ساعي البريد، يقوم بعملية "توزيع" بملاحظة أسماء إخوته وأخواته المكتوبة على الرسائل الواصلة وبعد ذلك يسلّم كل رسالة إلى صاحبها. تقوم آن بعملية "تجميع" من خلال أخذها الرسائل من إخوتها وأخواتها وتعطي البريد المجمّع إلى ساعي البريد.

الآن وقد فهمنا الدور المناط بعمليتي التجميع والتوزيع في طبقة النقل، دعنا نرى كيف يتم ذلك في واقع الأمر في مضيف على الشبكة. من المناقشة أعلاه يتبين أن التجميع في طبقة النقل يتطلّب: (1) أن يكون لكل مقبس معرّف يميزه، و(2) أن تتضمن كل قطعة بيانات حقولاً خاصة تشير إلى المقبس الذي ستسلّم القطعة له. هذه الحقول الخاصة والمبينة في الشكل 3-3 هي حقل رقم منفذ المصدر وحقل رقم

منفذ الوجهة (لقطع بيانات بروتوكولات TCP وUDP حقول أخرى أيضاً، كما سنرى في الأجزاء اللاحقة من هذا الفصل). يُمثّل رقم المنفذ بعدد يتكون من 16 بتاً، أي يتراوح من 0 إلى 65535. تخصص أرقام المنافذ من 0 إلى 1023 لاستخدام بروتوكولات التطبيقات المشهورة مثل HTTP (الذي يستخدم رقم المنفذ 80) وFTP (الذي يستخدم رقم المنفذ 20 و21). توجد قائمة بأرقام تلك المنافذ في RFC 1700 ومحدّثة في RFC 3232. عندما نطوّر تطبيقاً جديداً (كأحد التطبيقات التي طُوّرت في الأجزاء 2-7 و2-8)، يجب أن نخصص رقم منفذ للتطبيق.



الشكل 3-3 حقول أرقام المنافذ على كلٍّ من المصدر والوجهة بقطعة بيانات طبقة النقل.

لعله من الواضح الآن كيف تقوم طبقة النقل بإنجاز خدمة التوزيع (demultiplexing). يمكن تخصيص رقم منفذ لكل مقبس في المضيف، وعندما تصل قطعة بيانات إلى المضيف تقوم طبقة النقل بفحص رقم منفذ الوجهة في القطعة وتوجّه القطعة إلى المقبس المناظر، ومن ثم تمر بيانات القطعة من خلال المقبس إلى العملية المناظرة. كما سنرى، تلك هي الطريقة التي يتبعها بروتوكول UDP بشكلٍ أساسي. ومع ذلك فسنرى أيضاً أن التجميع والتوزيع في بروتوكول TCP هو أدق وألطف من ذلك.

التجميع والتوزيع اللاتوصيلي

تذكّر من الجزء 2-8 أنه بوسع برنامج جافا يتم تشغيله على مضيف أن ينشىء مقبساً باستخدام السطر التالي:

```
DatagramSocket mySocket = new DatagramSocket();
```

عند إنشاء مقبس UDP بهذه الطريقة، تقوم طبقة النقل تلقائياً بتخصيص رقم منفذ للمقبس. وبالتحديد تخصص طبقة النقل رقم منفذ في المدى من 1024 إلى 65535 يكون غير مستخدم حالياً من قِبَل أي منفذ UDP آخر على المضيف. وكطريقة بديلة يمكن لبرنامج جافا إنشاء مقبس باستخدام السطر التالي:

```
DatagramSocket mySocket = new DatagramSocket(19157);
```

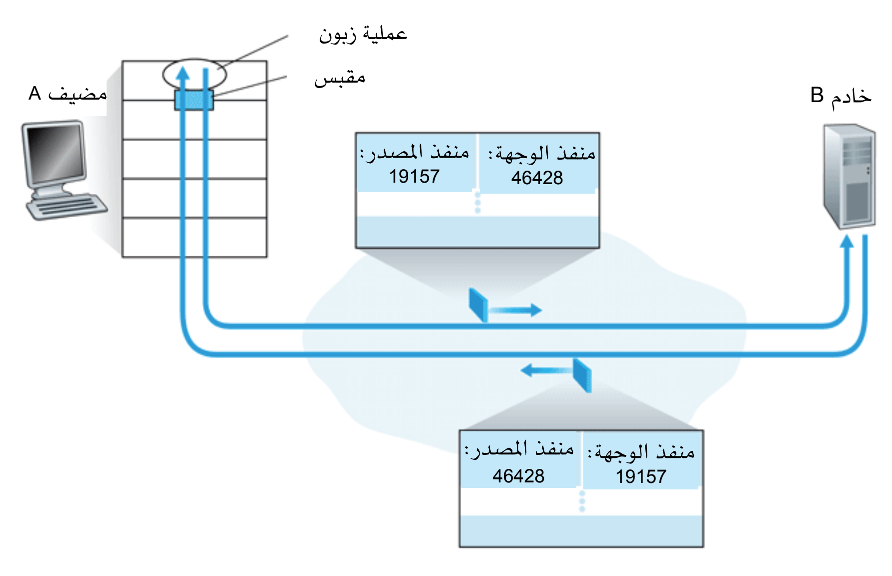
في هذه الحالة يخصص التطبيق رقم منفذ معين هو 19157 لمقبس الـ UDP. إذا كان مطور التطبيق الذي يكتب البرنامج ينجز جانب الخادم من بروتوكول مشهور، فعليه حينئذٍ تخصيص رقم المنفذ المناظر. عادةً ما يترك جانب الزبون من التطبيق لطبقة النقل القيام آليا (وبشفّافية) بتخصيص رقم المنفذ، بينما يقوم جانب الخادم من التطبيق بتخصيص رقم منفذ معين.

بعد تخصيص أرقام منافذ لمقابس UDP، يمكننا الآن وصف خدمتي التجميع والتوزيع بالضبط. افترض أن عملية في المضيف A تستخدم منفذ UDP رقم 19157 وتريد إرسال جزء من بيانات التطبيق إلى عملية تستخدم منفذ UDP رقم 46428 في المضيف B. تقوم طبقة النقل في المضيف A بتكوين قطعة طبقة نقل تتضمن بيانات التطبيق ورقم منفذ المصدر (19157) ورقم منفذ الوجهة (46428) وقيمتين أُخرَيَيْن (سنتناولهما لاحقاً، ولكنهما غير مهمّين للمناقشة الحالية). تقوم طبقة النقل بعد ذلك بدفع القطعة الناتجة إلى طبقة الشبكة. تغلف طبقة الشبكة القطعة في وحدة بيانات IP وتبذل أفضل جهد لتوصيل القطعة إلى مضيف الاستقبال. إذا وصلت القطعة إلى مضيف الاستقبال B، فإن طبقة النقل في المضيف تفحص رقم منفذ الوجهة في القطعة أي (46428) وتسلّم القطعة إلى المقبس المميّز بذلك الرقم.

لاحظ أن المضيف يمكن أن يكون عليه عدة عمليات تعمل في نفس الوقت، كل عملية لها مقبس UDP خاص بها ورقم منفذ مناظر. عند وصول قطع UDP من الشبكة يقوم المضيف B بتوجيه (توزيع) كل قطعة إلى المقبس المناظر بفحص رقم منفذ الوجهة الموجود على القطعة.

من المهم ملاحظة أن مقبس UDP يتم تمييزه بشكلٍ كامل بواسطة عنوان يضم عنوان IP للوجهة ورقم منفذ الوجهة، وكنتيجة لذلك إذا كان لقطعتي UDP عنوانا IP مختلفان للمصدر و/أو رقما منفذ مختلفان للمصدر، لكن لهما نفس عنوان IP ورقم المنفذ للوجهة، فإن القطعتين ستوجّهان إلى نفس عملية الوجهة النهائية عن طريق نفس مقبس تلك الوجهة.

قد تتساءل الآن وما فائدة رقم منفذ المصدر إذن؟ كما هو موضح في الشكل 3–4، في القطعة المتجهة من A إلى B يُستعمَل رقم منفذ المصدر كجزء من "عنوان العودة" – عندما يريد B إرسال قطعة إلى A، يأخذ منفذ الوجهة في القطعة المتجهة من B إلى A قيمته من قيمة منفذ المصدر في القطعة الواصلة من A إلى B (عنوان العودة الكامل هو عنوان IP للمضيف A ورقم منفذ المصدر).



الشكل 3-4 انعكاس أرقام منافذ المصدر والوجهة.

كمثال، تذكّر برنامج خادم UDP الذي تناولناه في الجزء 2-8. يستخدم الخادم في برنامج UDPServer.java طريقة لانتزاع رقم منفذ المصدر من القطعة التي يستلمها من الزبون، ثم يرسل قطعة جديدة إلى الزبون برقم منفذ المصدر المنتزع كرقم منفذ الوجهة في تلك القطعة الجديدة.

التجميع والتوزيع التوصيلي

لفهم وظيفة التوزيع (demultiplexing) في بروتوكول TCP يجدر بنا إلقاء نظرة فاحصة على مقابس TCP وطريقة إنشاء توصيلات TCP. هناك فرق دقيق بين مقبس UDP ومقبس TCP يكمن في أن مقبس TCP يتم تمييزه بعنوان رباعي: (عنوان IP للمصدر، ورقم منفذ المصدر، وعنوان IP للوجهة، ورقم منفذ الوجهة). وعليه فعندما تصل قطعة TCP من الشبكة إلى مضيف، يستعمل المضيف جميع تلك القيم الأربع لتوجيه (توزيع) القطعة إلى المقبس الملائم. بشكلٍ خاص وبالمقارنة مع UDP، فإنه عند وصول قطعتي TCP بعنواني IP مختلفين للمصدر أو رقمي منفذ مختلفين للمصدر سيوجّهان إلى مقبسين مختلفين (باستثناء قطعة TCP الأولى التي تحمل طلب إنشاء التوصيلة). لإلقاء مزيد من الضوء، دعنا نعيد النظر في مثال برمجة زبون/خادم TCP الذي استعرضناه في الجزء 2-7:

- لتطبيق الخادم في بروتوكول TCP "مقبس ترحيب" ينتظر من خلاله طلبات عمل التوصيلات من زبائن TCP على منفذ رقم 6789 (انظر الشكل 2-31)
- يقوم زبون TCP بتكوين قطعة إنشاء توصيلة (SYN) باستخدام السطر:

```
Socket clientSocket = new Socket ("serverHostName", 6789);
```

- طلب إنشاء توصيلة ماهو إلا قطعة TCP برقم منفذ وجهة = 6789 وفيها البت الخاص بإنشاء توصيلة في ترويسة TCP (سنتناولها في الجزء 3-5) له القيمة 1. تتضمن القطعة أيضاً رقم منفذ مصدر، والذي تم اختياره من قِبَل الزبون. يقوم السطر أعلاه أيضاً بإنشاء مقبس TCP لعملية الزبون حيث يمكن للبيانات أن تدخل وتغادر عملية الزبون من خلاله.

- عندما يستقبل نظام تشغيل الحاسب المضيف الذي يقوم بتشغيل عملية الخادم قطعة طلب إنشاء توصيلة والتي تحمل منفذ الوجهة رقم 6789، فإنه يقوم بتحديد عملية الخادم التي تنتظر قبول طلبات التوصيلات على منفذ رقم 6789. تقوم عملية الخادم بعد ذلك بإنشاء مقبس جديد:

```
Socket connectionSocket = welcomeSocket.accept();
```

- أيضاً تلاحظ طبقة النقل في الخادم القيم الأربع التالية في قطعة طلب التوصيلة: (1) رقم منفذ المصدر في القطعة، (2) عنوان IP لمضيف المصدر، (3) رقم منفذ الوجهة في القطعة، و(4) عنوان IP للمضيف الخاص بها. يتم تمييز مقبس التوصيلة الذي تم إنشاؤه حديثاً بتلك القيم الأربع؛ وكل القطع التي تصل بعد ذلك بنفس (منفذ المصدر، وعنوان IP للمصدر، ومنفذ الوجهة، وعنوان IP للوجهة) سيتم توزيعها إلى ذلك المقبس. الآن وقد تم تجهيز توصيلة TCP، يمكن للزبون والخادم أن يتبادلا إرسال البيانات بين بعضهما.

يمكن للمضيف الخادم دعم العديد من مقابس TCP في نفس الوقت، حيث يرتبط كل مقبس بعملية، ويُميّز كل مقبس بعنوانه الرباعي الخاص به. عندما تصل قطعة TCP إلى المضيف، تستعمل حقول العنوان الأربعة كلها (عنوان IP للمصدر، ومنفذ المصدر، وعنوان IP للوجهة، ومنفذ الوجهة) لتوجيه (توزيع) القطعة إلى المقبس الملائم.

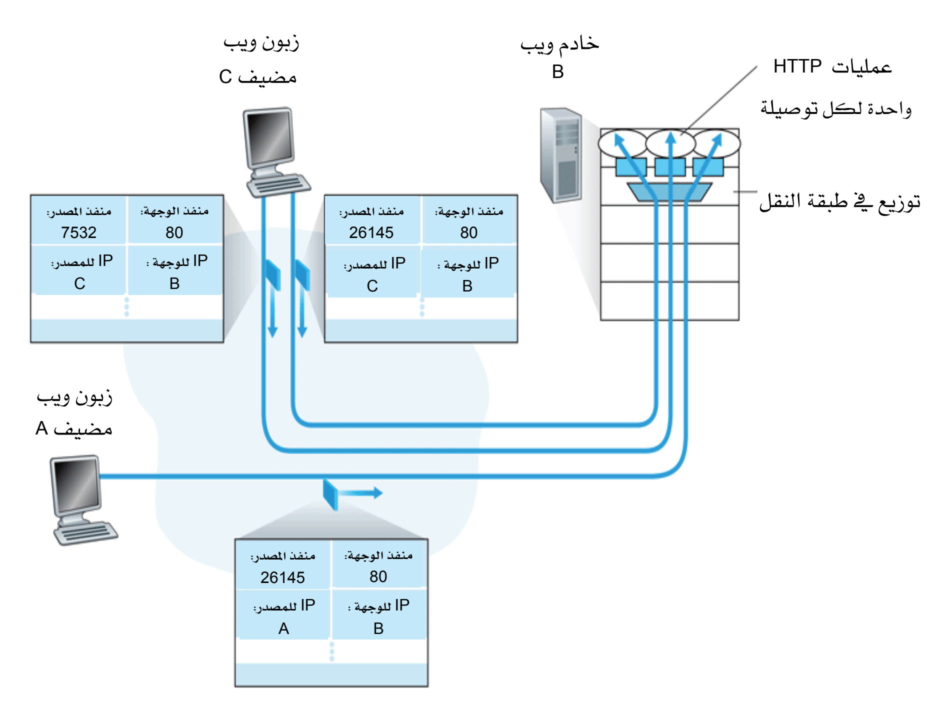
نبذة عن الأمن (Focus on Security)

مسح المنافذ:

رأينا أن عملية الخادم تنتظر بشغف ظهور منفذ مفتوح للاتصال من قِبَل زبون عن بعد. تُحجز بعض المنافذ للتطبيقات المشهورة (كخادمات الويب، وFTP، وDNS، وSMTP)، بينما تستخدم المنافذ الأخرى عادةً من قِبَل التطبيقات الأخرى المنتشرة (مثلاً: يُنصت خادم ميكروسوفت لقواعد البيانات SQL للطلبات على منفذ UDP رقم 1434). وهكذا إذا وجدنا منفذاً مفتوحاً على مضيف، يكون بوسعنا ربط ذلك المنفذ بتطبيق معين يجري تشغيله على المضيف. إن هذا مفيدٌ جداً لمدراء الأنظمة الذين غالباً ما يهتمون بمعرفة تطبيقات الشبكة المستخدمة على المضيفات في شبكاتهم. لكن المهاجمين – من منطلق "فحص المبنى قبل السطو عليه" – يريدون أيضاً معرفة المنافذ المفتوحة على المضيفات المستهدفة. فإذا وُجد مضيفٌ يشغّل تطبيقاً فيه نقطة ضعف أمنية معروفة، فإن المضيف يكون في ذلك الوقت جاهزاً للهجوم عليه (مثلاً يكون خادم SQL الذي يُنصت على منفذ 1434 عرضةً لفيض المخزن المؤقت buffer overflow، مما يسمح لمستخدم عن بعد بتشغيل أي برنامج اعتباطي على المضيف الضعيف، وهي نفس نقطة الضعف التي استغلتها دودة Slammer [CERT 2003-04]).

إن تحديد أي التطبيقات تُنصت على أي المنافذ في مضيف بعينه هو أمرٌ سهلٌ نسبياً، ففي الواقع هناك عدد من البرامج المتاحة للجميع على الإنترنت للقيام بذلك يطلق عليها اسم "ماسحات المنافذ" (port scanners)، ولعل أوسع تلك البرامج انتشاراً هو nmap، وهو متوفر مجاناً على الموقع http://insecure.org/nmap ومتضمن في معظم إصدارات نظام التشغيل لاينكس. يقوم برنامج nmap بمسح منافذ TCP بشكلٍ متسلسل بحثاً عن المنافذ التي تقبل توصيلات TCP. كما يقوم البرنامج بمسح منافذ UDP بشكلٍ متسلسل بحثاً عن المنافذ التي تستجيب لقطع UDP المُرسَلة. يعطي nmap في كلتا الحالتين قائمة تبين المنافذ المفتوحة والمغلقة وتلك التي تعذّر الوصول إليها. بوسع أي مضيف يقوم بتشغيل nmap محاولة مسح المنافذ على أي مضيف مستهدف في أي مكان على الإنترنت. سنزور برنامج nmap مرة أخرى في الجزء 3-5-6 عندما نناقش إدارة توصيلات بروتوكول TCP.

يبين الشكل 3-5 هذا الوضع، حيث يبدأ المضيف C جلستَي HTTP مع الخادم B، بينما يبدأ مضيف A جلسة HTTP واحدة مع الخادم B. لكلٍ من المضيفين A وC والخادم B عنوان IP الخاص به؛ لنفرض أن تلك العناوين هي A وC وB على التوالي. يخصص مضيف C رقمي منفذ مصدر مختلفين (26145 و7532) لتوصيلتي HTTP الخاصة به، ونظراً لأن المضيف A يختار أرقام منافذ المصدر بشكلٍ مستقل عن المضيف C، فربّما يخصّص رقم 26145 لمنفذ المصدر لتوصيلة HTTP الخاصة به، ولكن هذا لا يمثل مشكلة، فلا يزال بوسع الخادم B توزيع التوصيلتين بنفس رقم منفذ المصدر بشكلٍ صحيح، نظراً لأن التوصيلتين لهما عنوانا IP مختلفان للمصدر.



الشكل 3-5 زبونان يستخدمان نفس رقم منفذ الوجهة (80) للاتصال بنفس التطبيق على خادم ويب.

خادمات الويب وبروتوكول TCP

قبل أن ننتهي من هذه المناقشة، من المفيد ذكر بعض الملاحظات الإضافية حول خادمات الويب، والكيفية التي تستخدم بها أرقام المنافذ. خذ في الاعتبار مضيفاً يقوم بتشغيل خادم ويب Apache على منفذ 80. عندما يرسل الزبائن (مثلاً متصفّحات الويب) قطع البيانات إلى الخادم، سيكون لكل القطع الواصلة نفس منفذ الوجهة 80. على وجه الخصوص سيكون لكلٍّ من القطع الأولية الخاصة بإنشاء التوصيلة والقطع التي تحمل رسائل طلب HTTP نفس منفذ الوجهة 80. كما وضحنا أعلاه، يميّز الخادم القطع الواصلة من الزبائن المختلفة باستخدام عناوين IP للمصدر مع أرقام منافذ المصدر.

يبين الشكل 3-5 خادم ويب ينشىء عملية جديدة لكل توصيلة، ولكلٍ من هذه العمليات مقبس توصيلة خاص بها تستقبل من خلاله طلبات HTTP وترسل استجابات HTTP. ونذكر هنا أنه ليس هناك دائماً تناظر واحد لواحد بين كلٍّ من مقابس التوصيلات والعمليات، فغالباً ما تستخدم خادمات الويب الحديثة ذات مستوى الأداء العالي عملية واحدة فقط وتنشئ عملية فرعية بسيطة (thread) جديدة مع مقبس توصيلة جديد لكل طلب توصيلة جديد من الزبون. وإذا قمت بالإجابة على سؤال البرمجة الأول في الفصل الثانى، فإنك تكون قد طورت خادم ويب يؤدي ذلك بالضبط. قد يكون لمثل هذا الخادم في أي وقت من الأوقات العديد من مقابس التوصيلات المرتبطة بنفس العملية (بمعرّفات مختلفة).

إذا كان الزبون والخادم يستخدمان بروتوكول HTTP الدائم، فإنه طوال فترة التوصيلة الدائمة يتبادل الخادم والزبون رسائل HTTP عن طريق نفس مقبس الخادم. في حين إذا استخدم الخادم والزبون بروتوكول HTTP غير الدائم، فإن توصيلة TCP جديدة تُنشأ وتُغلق لكل طلب واستجابة، ويمكن أن يؤثر هذا الإنشاء والإغلاق المتكرّر للمقابس كثيراً على أداء خادم ويب مشغول (ورغم ذلك يوجد عدد من حيل نظام التشغيل التي يمكن استخدامها لتخفيف حدة تلك

المشكلة). وننصح القراء المهتمين بقضايا نظام التشغيل المتعلقة ببروتوكولات HTTP الدائمة وغير الدائمة بالاطلاع على [Nielsen 1997; Nahum 2002].

الآن وبعد أن انتهينا من مناقشة التجميع والتوزيع بطبقة النقل، ننتقل لمناقشة أحد بروتوكولات النقل على الإنترنت، ألا وهو UDP. وكما سنرى في الجزء التالي فإن بروتوكول UDP لا يضيف على بروتوكول طبقة الشبكة أكثر من خدمتي التجميع والتوزيع.

3-3 بروتوكول النقل اللاتوصيلي: UDP

سنلقي في هذا الجزء نظرةً فاحصةً على بروتوكول UDP، وكيف يعمل وماذا يعمل. وننصحك بمراجعة الجزء 2-1 الذي يتضمن نظرة عامة على نموذج خدمة UDP، والجزء 2-8 الذي يعالج برمجة المقابس باستخدام UDP.

لتحفيزك لمناقشتنا التالية حول UDP، افترض أنك مهتم بتصميم بروتوكول نقل بسيط يوفر الحد الأدنى من المتطلبات بلا رتوش. كيف ستبدأ في القيام بعملٍ كهذا؟ قد تفكر في البداية في استخدام بروتوكول نقل لا يضيف شيئاً سوى أخذ الرسائل من عملية التطبيق ودفعها مباشرة إلى طبقة الشبكة على جانب الإرسال؛ في حين على جانب الاستقبال، يرفع الرسائل التي تصله من طبقة الشبكة مباشرة إلى عملية التطبيق. غير أننا – كما تعلّمنا في الجزء السابق – يجب أن نفعل أكثر قليلاً من لا شيء!، فعلى أقل تقدير يجب أن توفر طبقة النقل خدمة تجميع وتوزيع لكي تنقل البيانات بين طبقة الشبكة والعملية الملائمة في طبقة التطبيقات.

يقوم بروتوكول النقل UDP والمعرّف في RFC 768 تقريباً بالحد الأدنى فقط المطلوب من بروتوكول نقل. باستثناء وظيفتي التجميع والتوزيع وبعض التدقيق الخفيف لاكتشاف الأخطاء، لا يضيف UDP شيئاً يُذكر إلى بروتوكول IP. في الحقيقة إذا اختار مطور التطبيقات البروتوكول UDP بدلاً من TCP، فإن التطبيق سيتعامل مباشرةً تقريباً مع بروتوكول IP. يأخذ UDP الرسائل من عملية التطبيق، ويضيف حقلين لرقم منفذ المصدر ورقم منفذ الوجهة، كما يضيف حقلين صغيرين

آخرين، ويدفع بالقطعة الناتجة إلى طبقة الشبكة. تغلّف طبقة الشبكة قطعة طبقة النقل للحصول على وحدة بيانات IP، ثم بعد ذلك تبذل الطبقة أفضل جهد لتوصيل القطعة إلى مضيف الاستقبال. إذا وصلت القطعة إلى مضيف الاستقبال، يستعمل UDP رقم منفذ الوجهة لتوصيل بيانات القطعة إلى عملية التطبيق الملائمة. لاحظ أنه مع بروتوكول UDP ليس هناك إجراءات مصافحة (handshaking) بين كيانات طبقتي النقل المُرسَلة والمُستقبِلة قبل إرسال قطعة. ولهذا السبب يعرف بروتوكول UDP بأنه بروتوكول غير توصيلي (connectionless).

يُعتبر بروتوكول DNS (نظام أسماء النطاقات) مثالاً لبروتوكولات طبقة التطبيقات التي تستخدم UDP عادةً. فعندما يريد DNS في مضيف عمل استفسار، فإنه يُنشىء رسالة استفسار DNS ويدفع بها إلى UDP. ويضيف UDP على جانب المضيف حقول الترويسة إلى الرسالة ويدفع بالقطعة الناتجة إلى طبقة الشبكة وذلك بدون أي إجراءات مصافحة مع كيان UDP الذي يجري تشغيله على النظام الطرفي في الوجهة. تغلّف طبقة الشبكة قطعة UDP لتكوّن وحدة بيانات طبقة الشبكة (datagram) وترسلها إلى خادم أسماء النطاقات. ينتظر DNS في المضيف المستفسر إجابةً على استفساره. إذا لم يتلق إجابة (ربما لأن الشبكة التحتية فقدت الاستفسار أو الإجابة)، فإما أن يحاول إرسال الاستفسار إلى خادم آخر للأسماء، أو يخبر التطبيق الذي يستخدمه بأنّه لا يستطيع الحصول على إجابة.

قد تتساءل الآن: وإذا كان الأمر كذلك فلماذا يختار مطور للتطبيقات تطوير تطبيقه على بروتوكول UDP بدلاً من بروتوكول TCP؟ أليس بروتوكول TCP مفضلاً على الدوام، حيث إنه يوفر خدمة موثوقة لنقل البيانات على العكس من UDP؟ الجواب لا، فالعديد من التطبيقات يلائمها بروتوكول UDP أكثر للأسباب التالية:

- التحكم الأدق من قبل التطبيق في البيانات التي تُرسَل ومتى تُرسَل تبعاً لبروتوكول UDP. فبمجرد أن تمرر عملية التطبيق البيانات إلى UDP، يقوم البروتوكول بوضع البيانات داخل قطعة UDP ودفعها على الفور إلى طبقة الشبكة. وعلى العكس من ذلك، يتضمن بروتوكول TCP آلية للتحكم

في الازدحام تخنق مُرسِل طبقة النقل عندما تصبح واحدة أو أكثر من الوصلات بين مضيفي المصدر والوجهة النهائية مزدحمة بشكلٍ مُفرط. من ناحية أخرى يواصل بروتوكول TCP إعادة إرسال قطعة البيانات المرة تلو الأخرى إلى أن يصله إشعار استلام (acknowledgment) من الوجهة بوصول القطعة، بغض النظر عن الوقت الذي يستغرقه التوصيل الموثوق للقطعة. إلا أن التطبيقات الفورية غالباً ما تتطلب ضمان معدل أدنى لإرسال البيانات، ولا تحبذ تأخير إرسال قطع البيانات كثيراً، حيث يمكنها أن تتحمل بعض الفقد في البيانات، لذا فإن نموذج الخدمة الذي يوفره بروتوكول TCP لا يلائم حاجة تلك التطبيقات بشكلٍ جيد كما سنبين لاحقاً، ويمكن لهذه التطبيقات استخدام بروتوكول UDP وتطوير أي وظائف إضافية مطلوبة علاوة على خدمة UDP كجزء من التطبيق.

- عدم الحاجة لإنشاء توصيلة: كما سنرى لاحقاً يستخدم بروتوكول TCP آلية ثلاثية للمصافحة قبل البدء في نقل البيانات، لكن على العكس من ذلك ينطلق بروتوكول UDP في نقل البيانات على الفور دون أي تمهيدات رسمية، ولذا فإن UDP لا يعاني من أي تأخير لإنشاء توصيلة. لعل هذا هو السبب الرئيس الذي بسببه يستخدم DNS بروتوكول UDP وليس TCP. إن DNS سيكون أبطأ بكثير إذا استخدم TCP. في المقابل يستخدم HTTP بروتوكول TCP بدلاً من UDP لأن الموثوقية شيء مهم فيما يتعلق بصفحات الويب التي تتضمن نصوصاً. لكن – كما استعرضنا سريعاً في الجزء 2-2 – يُعتبر التأخير بسبب إنشاء توصيلة في HTTP من عوامل التأخير الأساسية عند تنزيل صفحات الويب.
- عدم الاكتراث بحالة التوصيلة: يحتفظ بروتوكول TCP بـ"حالة التوصيلة" في الأنظمة الطرفية. تتضمن حالة التوصيلة هذه المخازن المؤقتة للإرسال والاستقبال، ومتغيرات التحكم في الازدحام، ومتغيرات الأرقام المتسلسلة، وأرقام إشعارات الاستلام. سنرى في الجزء 3–5 أن معلومات حالة التوصيلة هذه لازمة لتحقيق خدمة نقل موثوقة للبيانات مع التحكم في الازدحام في

بروتوكول TCP. في المقابل لا يحتفظ UDP بحالة توصيلة ولا يتتبع أياً من تلك المتغيرات، لذا فإن خادماً مخصصاً لتطبيق معين يمكنه عادةً دعم عدد أكبر بكثير من الزبائن عندما يستخدم التطبيق بروتوكول UDP بدلاً من TCP.

- تقليل الأعباء الإضافية بسبب ترويسة قطعة البيانات: لكل قطعة بيانات في بروتوكول TCP ترويسة تتكون من 20 بايتاً، في حين أن العبء الإضافي في حالة بروتوكول UDP يقتصر على 8 بايتات فقط.

يبين الشكل 3-6 تطبيقات الإنترنت المعروفة وبروتوكولات النقل التي تستعملها. كما نتوقّع فإن تطبيقات البريد الإلكتروني، والدخول على الحاسبات عن بعد، وتصفح الويب، ونقل الملفات، كلها تستخدم بروتوكول TCP – فكل تلك التطبيقات تحتاج لخدمة النقل الموثوق التي يوفرها TCP. ومع ذلك فالعديد من التطبيقات المهمة تستخدم UDP بدلاً من TCP. يُستخدم UDP على سبيل المثال لتحديث جداول التوجيه من قبل بروتوكول RIP (انظر الجزء 4-6-1)، وذلك نظراً لأن تحديثات RIP تتم بشكلٍ دوري (كلّ خمس دقائق عادةً)، وبالتالي فإن التحديثات التي تفقد في الطريق ستستبدل بتحديثات تالية، مما يجعل التحديثات المفقودة المنتهي تاريخها عديمة الفائدة. كما يُستخدم UDP أيضاً لتشغيل تطبيقات إدارة الشبكة من خلال بروتوكول SNMP (انظر الفصل التاسع). ويرجع سبب تفضيل UDP على TCP في هذه الحالة إلى أن تطبيقات إدارة الشبكة يجب تشغيلها عادةً والشبكة في حالة مجهَدة ومزدحمة، وهو بالضبط الوقت الذي يصعب فيه تحقيق نقل موثوق للبيانات. كما ذكرنا أعلاه، يستخدم UDP أيضاً من قِبَل بروتوكول DNS، وبذلك يتفادى الأخير تأخيرات إنشاء التوصيلات فيما لو استخدم TCP.

التطبيق	بروتوكول طبقة التطبيقات	بروتوكول طبقة النقل التحتي
البريد الإلكتروني	SMTP	TCP
للدخول على الحاسبات عن بعد	Telnet	TCP
الويب	HTTP	TCP
نقل الملفّات	FTP	TCP
نقل الملفّات عن بعد	NFS	عادةً UDP
عرض مواد الوسائط المتعددة	مملوكة	UDP أو TCP
هاتف الإنترنت	مملوكة	UDP أو TCP
هاتف الإنترنت	SNMP	عادةً UDP
بروتوكول التوجيه	RIP	عادةً UDP
تحويل أسماء النطاقات	DNS	عادةً UDP

الشكل 3-6 تطبيقات الإنترنت المشهورة وبروتوكولات طبقة النقل التحتية المستخدمة معها.

كما يظهر من الشكل 3-6، يُستخدَم كلٌّ من البروتوكولين UDP وTCP حالياً من قِبَل تطبيقات الوسائط المتعددة، كهاتف الإنترنت، والمؤتمرات الفورية عبر الفيديو، وتشغيل تسجيلات الصوت والفيديو المخزّنة. سنلقي نظرةً فاحصةً على هذه التطبيقات في الفصل السابع. فقط نذكر هنا أن كل تلك التطبيقات يمكن أن تتحمّل قدراً قليلاً من فقد الرزم، ولذا فإن النقل الموثوق للبيانات ليس ضرورياً جداً لنجاح التطبيق. علاوة على ذلك فإن التطبيقات الفورية، كهاتف الانترنت والمؤتمرات عبر الفيديو، تستجيب بشكلٍ سيئ جداً لوسائل التحكم في الازدحام التي يفرضها بروتوكول TCP. لهذه الأسباب قد يختار مطوّرو تطبيقات الوسائط المتعددة تشغيل تطبيقاتهم على UDP بدلاً من TCP. ومع ذلك، فإن TCP يُستخدم الآن على نحو متزايد لنقل مواد عرض الوسائط المتعددة. على سبيل المثال، وجد [Sripanidkulchai 2004] أن حوالي 75٪ من تطبيقات الفيديو للعرض الحي والعرض حسب الطلب تستخدم TCP. عندما تكون نسب فقد الرزم المسموح بها منخفضة، وعندما تحجب شبكات بعض الهيئات حركة مرور البيانات التي تستخدم

بروتوكول UDP لأسباب أمنية (انظر الفصل الثامن)، يصبح TCP بديلاً لا عوض عنه لنقل مادة عرض الوسائط المتعددة.

رغم أن استخدام تطبيقات الوسائط المتعددة لبروتوكول UDP منتشر اليوم، إلا أن الأمر لايزال محل خلاف. كما ذكرنا أعلاه لا يتضمن بروتوكول UDP تحكماً في الازدحام، ولكن التحكم في الازدحام مطلوب لمنع الشبكة من دخول حالة اختناق يتعذر فيها القيام بأي عمل مفيد. إذا استخدم كل شخص العرض المستمر لأفلام الفيديو بمعدل عالٍ لإرسال البيانات بدون أي تحكم في الازدحام، فسيغمر فيض كبير من الرزم موجّهات الشبكة بحيث يتمكن عدد قليل جداً من رزم UDP من قطع المسار من المصدر إلى الوجهة النهائية بنجاح. وعلاوةً على ذلك فإن نسب الفقد العالية للرزم بسبب عدم التحكم في معدلات الإرسال من قِبَل مُرسِلي UDP ستؤدي بمُرسِلي TCP (والذين، كما سنرى، يخفضون معدلات إرسالهم لمواجهة الازدحام) إلى تقليل معدلاتهم بشكلٍ كبير. وهكذا فإن عدم التحكم في الازدحام في UDP يمكن أن تؤدي إلى نسب فقد عالية بين مُرسِلي ومستقبلي UDP، بالإضافة إلى ازدحام جلسات TCP؛ وتلك مشكلة خطيرة فعلاً [Floyd 1999]. اقترح العديد من الباحثين آليات جديدة لإلزام كل مصادر البيانات، بما في ذلك مصادر UDP، بالقيام بتحكم يتواءم مع الازدحام [Mahdavi 1997; Floyd 2000; Kohler 2006; RFC 4340].

قبل مناقشة صيغة قطعة بيانات بروتوكول UDP، نذكر هنا أنه يمكن لتطبيقٍ ما تحقيق نقلٍ موثوقٍ للبيانات مع استعمال بروتوكول UDP، وذلك ببناء تلك الموثوقية في التطبيق نفسه (على سبيل المثال بإضافة إشعارات الاستلام وآليات إعادة الإرسال كالتي سندرسها في الجزء القادم)، غير أن هذه مهمّة ليست بالبسيطة، ويمكنها أن تشغل مطور التطبيقات بتنقيح وتصحيح برامجه لفترة طويلة. ومع ذلك فبناء الموثوقية مباشرةً في التطبيق يسمح للتطبيق بـ"الاحتفاظ بكعكته وأكلها أيضاً"، بمعنى أن عمليات التطبيقات يمكن أن تتصل بشكلٍ موثوق بدون أن تخضع للقيود على معدلات الإرسال التي تفرضها آليات التحكم في الازدحام ضمن بروتوكول TCP.

3-3-1 صيغة قطعة بيانات UDP

تم توصيف صيغة قطعة بيانات بروتوكول UDP في RFC 768 كما هو مبين في الشكل 3-7. تحتل بيانات التطبيق حقل البيانات (الحمل الآجر) في قطعة UDP. على سبيل المثال في حالة DNS يتضمن حقل البيانات إما رسالة استفسار أو رسالة رد. أما في تطبيق عرض صوتي فتملأ عينات من المادة الصوتية حقل البيانات. تتكون ترويسة UDP من أربعة حقول يتألف كلٌّ منها من بايتين. كما ذكرنا في القسم السابق تسمح أرقام المنافذ لمضيف الوجهة بتمرير بيانات التطبيق إلى العملية الملائمة التي يجري تنفيذها على النظام الطرفي للوجهة (أي لأداء وظيفة التوزيع demultiplexing). يستخدم مضيف الاستقبال حقل "المجموع التدقيقي" (checksum) لاكتشاف ما إذا كانت هناك أخطاء قد طرأت على القطعة أثناء انتقالها من مضيف المصدر. وفي الواقع يُحسب "المجموع التدقيقي" أيضاً على عدة حقول في ترويسة IP بالإضافة إلى قطعة UDP. لكنّنا سنهمل هذه التفاصيل لكي نتمكن من "رؤية الغابة من خلال الأشجار"، وسنناقش طريقة حساب المجموع التدقيقي في الجزء التالي، كما سنتناول المبادئ الأساسية لكشف الخطأ في الجزء 5-2. يحدد حقل الطول طول قطعة UDP الكلي بالبايتات بما في ذلك حقل الترويسة.

الشكل 3-7 صيغة قطعة بيانات UDP.

3-3-2 حقل المجموع التدقيقي بقطعة بيانات UDP

يُستخدم حقل المجموع التدقيقي بقطعة بيانات UDP لاكتشاف الأخطاء. بمعنى أنه يُستعمل لمعرفة ما إذا كانت البتات في القطعة قد طرأ عليها أي تغيير أثناء انتقالها من المصدر إلى الوجهة النهائية – على سبيل المثال بسبب الشوشرة في الوصلات أو عند التخزين المؤقت في الموجّهات. يقوم بروتوكول UDP في ناحية المُرسِل بحساب المكمّل للواحد (1's complement) لناتج جمع كل الكلمات (بطول 16 بتاً) الموجودة في القطعة، مع تدوير أي فيض (overflow) يحدث في الخانة الأخيرة وإضافته إلى الخانة الأولى. توضع النتيجة في حقل المجموع التدقيقي لقطعة UDP.

وفيما يلي مثال بسيط لتوضيح حساب المجموع التدقيقي، لكن يمكنك الإطلاع على تفاصيل طرق عالية الكفاءة لإجراء هذه العملية في RFC 1071 ومعلومات عن أدائها على بيانات حقيقية في [Stone 1998; Stone 2000]. افترض أن لدينا الكلمات الثلاث التالية بالقطعة (كل منها يتألف من 16 بتاً):

```
0110011001100000
0101010101010101
1000111100001100
```

حاصل جمع أول كلمتين من هذه الكلمات الثلاث هو:

```
 0110011001100000
 0101010101010101
 ----------------
 1011101110110101
```

بإضافة الكلمة الثالثة نحصل على:

```
 1011101110110101
 1000111100001100
 ----------------
 0100101011000001
               +1
 ----------------
 0100101011000010
```

لاحظ حدوث فيض في نهاية عملية الجمع الأخيرة، وقد تم تدويره وإضافته للخانة الأولى. نحصل على مكمل الواحد للناتج بتحويل كل 0 إلى 1 وكل 1 إلى 0. وهكذا فإن مكمل الواحد للمجموع 0100101011000010 هو 1011010100111101، والذي يوضع في حقل المجموع التدقيقي. في مضيف الوجهة يتم جمع كل الكلمات الأربع بالقطعة (في هذا المثال) بما في ذلك قيمة المجموع التدقيقي. إذا لم تحدث أي أخطاء في بتات القطعة، فمن الواضح أن المجموع سيكون 1111111111111111. أما إذا كان أحد البتات في الناتج له القيمة 0، فسندرك أن خطأ أو أكثر قد طرأ على القطعة أثناء رحلتها من المصدر إلى الوجهة النهائية.

قد تتساءل لماذا يتضمن بروتوكول UDP إمكانيات لاكتشاف الأخطاء أساساً، حيث إن الكثير من بروتوكولات طبقة ربط البيانات (بما في ذلك بروتوكول الإيثرنت الشهير) يوفر تلك الإمكانيات أيضاً. يكمن السبب في أنه ليس هناك ما يضمن أن كل الوصلات بين المصدر والوجهة تقوم بالكشف عن الأخطاء؛ بمعنى أن إحدى الوصلات قد تستخدم بروتوكول طبقة ربط بيانات لا يقوم بذلك. علاوة على ذلك فحتى إذا انتقلت القطع عبر الوصلة بشكلٍ صحيح، فمن المحتمل حدوث أخطاء في القطعة أثناء تخزينها في ذاكرة أحد الموجّهات. فإذا كان كلٌّ من موثوقية نقل البيانات على الوصلات واكتشاف أخطاء التخزين في ذاكرات الموجّهات غير مضمون، فلا غرابة إذن في أن يوفر بروتوكول UDP إمكانية اكتشاف الأخطاء في طبقة النقل – على أساس من طرف إلى طرف – إذا كان ذلك مطلوباً. يعتبر هذا مثالاً للمبدأ المشهور في تصميم الأنظمة "من طرف إلى طرف" [Saltzer 1984]، والذي يقول بأنه لمّا كان من الضروري القيام ببعض المهام على أساس من طرف إلى طرف (كاكتشاف الأخطاء في هذه الحالة) فإن "الوظائف المتعلقة التي تنفّذ في المستويات الأدنى قد تكون زائدة أو ذات قيمة ضئيلة بالمقارنة بكلفة أدائها في المستوى الأعلى".

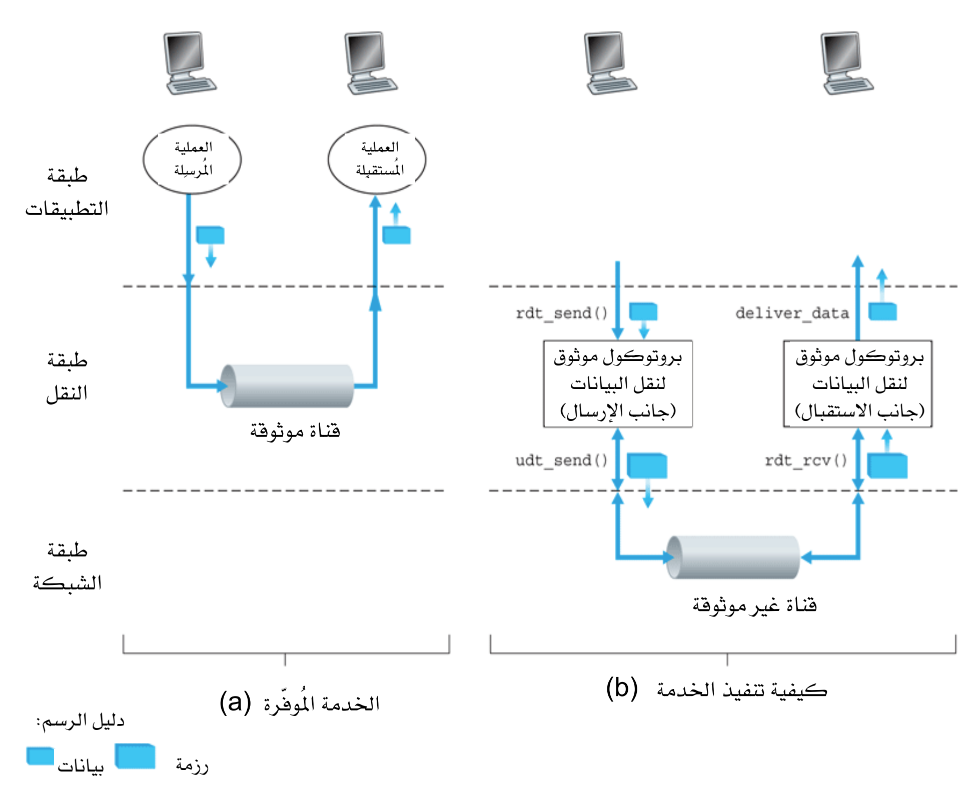
نظراً لأنه يفترض تشغيل IP فوق أي بروتوكول في الطبقة الثانية، فمن المفيد لطبقة النقل توفير إمكانيات اكتشاف الأخطاء كإجراء وقائي. ورغم أن بروتوكول UDP يوفر إمكانية التدقيق لاكتشاف الأخطاء، إلا أنه لا يتخذ أي إجراء لتصحيح أي خطأ يتم اكتشافه، فبعض تطبيقات UDP ببساطة تهمل قطع البيانات التي فيها خطأ، بينما تمرر التطبيقات الأخرى تلك القطع إلى التطبيق مع التنبيه لتلك الأخطاء.

وهكذا نصل إلى نهاية استعراضنا لبروتوكول UDP، وسنرى قريبا أن بروتوكول TCP يوفر لتطبيقاته نقلاً موثوقاً للبيانات بالإضافة إلى الخدمات الأخرى التي لا يوفرها بروتوكول UDP. من الطبيعي أيضاً أن يكون TCP أكثر تعقيداً عن UDP. ولكن قبل أن نتناول بروتوكول TCP بالتفصيل، سيكون من المفيد أن نرجع خطوةً للوراء لنناقش المبادئ الأساسية للنقل الموثوق للبيانات.

3-4 أساسيات النقل الموثوق للبيانات

في هذا الجزء سنتناول النقل الموثوق للبيانات في سياق عام. هذا أمر ملائم، حيث إن تحقيق نقل موثوق للبيانات أمرٌ مطلوب ليس فقط في طبقة النقل، ولكن أيضاً في طبقة ربط البيانات وطبقة التطبيقات، وعليه فإن المشكلة بهذا العموم لها أهميتها الأساسية في مجال الشبكات. في الواقع إذا طُلب من شخص سرد قائمة بأهم عشر مشاكل أساسية في مجال ربط الشبكات، فقد تكون تلك المشكلة مرشحة لشغل المركز الأول. في الجزء القادم سندرس بروتوكول التحكم في الإرسال TCP، وسنرى بشكلٍ خاص أن هذا البروتوكول يستثمر العديد من المبادئ التي نحن بصدد تناولها الآن.

يوضح الشكل 3-8 الإطار العام لدراستنا للنقل الموثوق للبيانات. يتمثل نموذج الخدمة التي يتم توفيرها للطبقات الأعلى في قناةٍ موثوقة يمكن نقل البيانات عبرها. في تلك القناة الموثوقة لا تتعرض بتات البيانات للتغير (0 يتحول إلى 1 أو العكس) ولا تفقد وتصل بنفس الترتيب الذي أُرسلت به. هذا بالضبط هو "نموذج الخدمة" الذي يوفره بروتوكول TCP لتطبيقات الإنترنت التي تستخدمه.



الشكل 3-8 النقل الموثوق للبيانات: (a) نموذج الخدمة، (b) كيفية تنفيذ الخدمة.

تتلخص مسؤولية بروتوكول النقل الموثوق للبيانات في تنفيذ هذا النموذج المذكور للخدمة. مما يجعل هذه المهمة صعبة أن الطبقة تحت بروتوكول النقل الموثوق للبيانات قد تكون غير موثوقة في واقع الأمر. فمثلاً بروتوكول TCP للنقل الموثوق للبيانات يعمل فوق بروتوكول IP في طبقة الشبكة والذي لا يوفر موثوقية للنقل من طرف إلى طرف. وبشكلٍ أكثر عموماً، قد تتألف الطبقة تحت النقطتين الطرفيتين اللتين تتصلان بشكلٍ موثوق من وصلة مادية واحدة فقط (كما في حالة بروتوكول وصلة البيانات)، أو من شبكة عالمية (كما في حالة بروتوكول طبقة النقل). وعلى أية حال يكفي من أجل المناقشة هنا أن نعتبر تلك الطبقة السفلى ببساطة كقناة غير موثوقة من نقطة لنقطة.

في هذا الجزء سنطوّر تدريجياً جوانب الإرسال والاستقبال لبروتوكول نقل موثوق للبيانات، مع الأخذ في الاعتبار نماذج معقدة أكثر فأكثر للقناة التحتية. يبين الشكل 3-8 (b) الواجهات (interfaces) الخاصة ببروتوكولنا للنقل الموثوق للبيانات. سيتم طلب جانب الإرسال في البروتوكول من أعلى عن طريق نداء لـ rdt_send() والذي يقوم أيضاً بتمرير البيانات المطلوب توصيلها إلى الطبقة الأعلى في جانب الاستقبال. (ترمز rdt هنا لـ reliable data transfer protocol (أي بروتوكول النقل الموثوق للبيانات) بينما تشير send_ إلى أن النداء موجّه لجانب الإرسال من البروتوكول. إنّ الخطوة الأولى في تطوير أي بروتوكول تكمن في اختيار اسم جيد له!). على جانب الاستقبال سيتم النداء لـ rdt_rcv() عندما تصل رزمة بيانات من جانب الاستلام في القناة. عندما يريد بروتوكول rdt توصيل البيانات إلى الطبقة الأعلى، سيقوم بذلك عن طريق نداء لـ deliver_data(). سنستخدم هنا "رزمة" بدلاً من "قطعة" الخاصة بطبقة النقل. نظراً لأن النظرية التي نقوم بتطويرها في هذا الجزء تنطبق على شبكات الحاسب بصفة عامة وليس فقط على طبقة النقل في الإنترنت، فإن التعبير العام "رزمة" ربما يكون أكثر ملاءمة هنا.

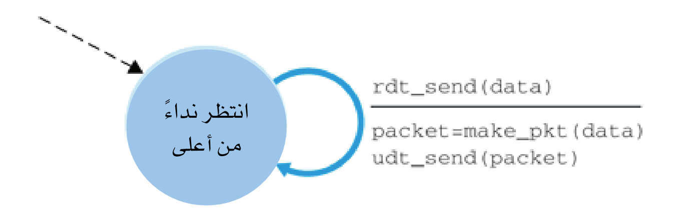
في هذا الجزء سنتناول حالة نقل البيانات في اتجاه واحد فقط: من جانب الإرسال إلى جانب الاستقبال. رغم أن حالة النقل الموثوق للبيانات في اتجاهين (أي بشكلٍ كامل الازدواج full-duplex) ليست أصعب كثيراً من حيث المفهوم إلا أنها أعقد بكثير في الشرح. ورغم أننا سنأخذ في الاعتبار نقل البيانات في اتجاه واحد فقط، إلا أنه من المهم ملاحظة أن جانبي الإرسال والاستقبال في بروتوكولنا سيحتاجان مع ذلك لإرسال رزم في كلا الاتجاهين كما يبين الشكل 3-8. سنرى بعد قليل أن جانبي الإرسال والاستقبال في البروتوكول rdt سيحتاجان إلى تبادل رزم التحكم ذهاباً وإياباً بالإضافة إلى تبادل الرزم التي تحتوي على بيانات مطلوب نقلها. يقوم كلا الجانبين من rdt بإرسال الرزم إلى الجانب الآخر بواسطة نداء لـ udt_send (حيث ترمز udt لنقل بيانات غير موثوق).

3-4-1 بناء بروتوكول للنقل الموثوق للبيانات

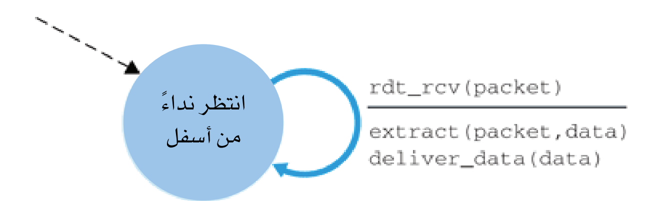
نتدرج الآن عبر سلسلة من البروتوكولات، كلٌّ منها أكثر تعقيداً من سابقه، لنصل في النهاية إلى بروتوكول سليمٍ خالٍ من العيوب للنقل الموثوق للبيانات.

نقل موثوق للبيانات عبر قناة موثوقة تماماً : rdt1.0

سنتناول أولاً الحالة الأسهل التي تكون فيها القناة التحتية موثوقة تماماً. في هذه الحالة يكون البروتوكول نفسه — والذي سنطلق عليه rdt1.0 — بديهياً. يبين الشكل 3-9 تعريفات آلة الأوضاع المحدودة (finite state machine (FSM)) لمُرسِل ومُستقبِل البروتوكول rdt1.0. تعرّف FSM في الشكل 3-9 (a) عملية المُرسِل، بينما تعرّف FSM في الشكل 3-9 (b) عملية المُستقبِل. من المهم ملاحظة أن هناك FSM مستقلة لكلٍ من المُرسِل والمُستقبِل. آلة FSM الخاصة بكلٍ من المُرسِل والمُستقبِل في الشكل 3-9 لها وضع واحد فقط. تشير الأسهم في وصف FSM إلى انتقال البروتوكول من وضع لآخر (نظراً لأن كل FSM في الشكل 3-9 لها وضع واحد فقط، فإن الانتقال يكون بالضرورة من ذلك الوضع إلى نفسه؛ سنرى مخططات حالات أكثر تعقيداً بعد قليل). يُبيّن الحدث الذي يسبّب الانتقال فوق الخط الذي يمثل الانتقال، كما تُبيّن الإجراءات التي تُتخذ عند حدوث الحدث تحت ذلك الخط. عندما لا يُتخذ أي إجراء عند وقوع حدث سنضع الرمز Λ تحت الخط، وعندما لا يقع حدث ولكن يتم اتخاذ إجراء سنضع الرمز Λ فوق الخط وذلك للدلالة بوضوح على عدم وجود إجراء أو حدث. يُبيّن وضع البداية لآلة FSM بسهم متقطع. ورغم أن كلاًّ من آلات FSM في الشكل 3-9 لها وضع واحد فقط، فإننا سنرى بعد قليل آلات FSM لها عدة أوضاع، لذا سيكون من المهم تمييز وضع البداية لكل آلة FSM.



(a) بروتوكول rdt1.0: جانب الإرسال



(b) بروتوكول rdt1.0: جانب الاستقبال

الشكل 3-9 بروتوكول rdt1.0 لقناة موثوقة تماماً.

يقبل جانب الإرسال من بروتوكول rdt ببساطة البيانات من الطبقة الأعلى عن طريق الحدث rdt_send(data)، ومن ثم يكوّن رزمة تتضمن البيانات (بواسطة الإجراء make_pkt(data)، وبعد ذلك يرسل الرزمة إلى القناة. عملياً ينشأ الحدث rdt_send(data) كنتيجة لنداء إجراء (على سبيل المثال rdt_send() صادر عن تطبيق في الطبقة الأعلى.

على جانب الاستقبال، يستلم rdt رزمة من القناة التحتية عن طريق الحدث rdt_rcv(packet)، ثم يأخذ البيانات من الرزمة بواسطة الإجراء extract (packet, data)، ومن ثم تعبر البيانات إلى الطبقة الأعلى عن طريق الإجراء deliver_data (data). عملياً ينشأ الحدث rdt_rcv(packet) نتيجة لنداء إجراء (على سبيل المثال rdt_rcv() من بروتوكول الطبقة السفلى.

في هذا البروتوكول البسيط ليس هناك اختلاف بين وحدة البيانات والرزمة. أيضاً يكون كل تدفق للرزم من المُرسِل إلى المُستقبِل؛ فعند استخدام قناة اتصال موثوقة تماماً لا توجد حاجة لجانب المُستقبِل لتزويد جانب المُرسِل بأي تعقيبات أو تغذية مرتدة، حيث لا يمكن حدوث خطأ! لاحظ أننا افترضنا أيضاً أن المُستقبِل يستطيع استلام البيانات بنفس السرعة التي يرسلها بها المُرسِل. وعليه فليست هناك حاجة للمُستقبِل لأن يطلب من المُرسِل التباطؤ!

نقل موثوق للبيانات عبر قناةٍ بأخطاء في البتات: rdt2.0

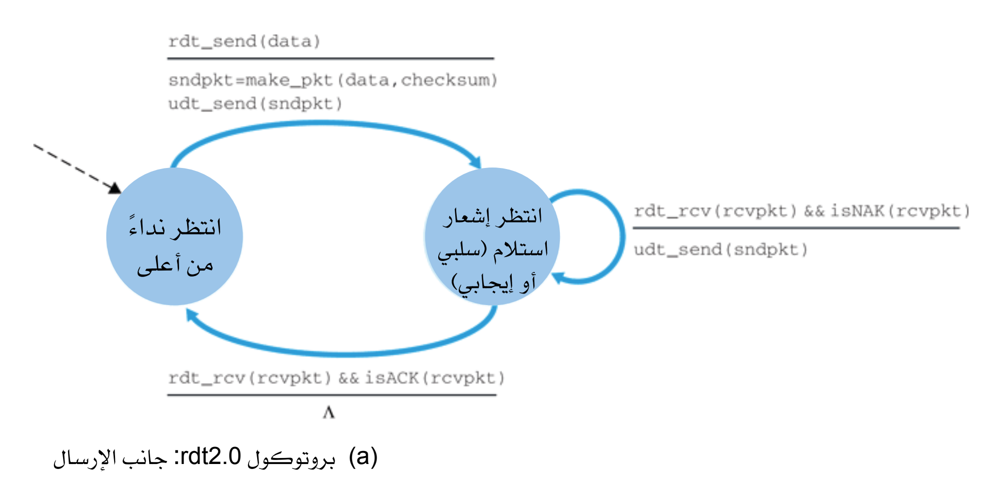
النموذج الأكثر واقعية للقناة التحتية هو ذلك الذي تتعرض فيه بتات الرزمة للخطأ. تحدث تلك الأخطاء في البتات عادةً في المكوّنات المادية للشبكة أثناء إرسال أو نقل الرزمة المُرسَلة، أو أثناء حفظ الرزمة في مخزن مؤقت. سنواصل الافتراض أيضاً بأن كل الرزم المُرسَلة يتم استلامها (رغم أن بتاتها قد يعتريها الخطأ).

قبل تطوير بروتوكول للاتصال الموثوق عبر مثل تلك القناة، لنأخذ في الاعتبار كيف يمكن أن يتعامل البشر مع هذه الحالة. تأمل كيف تقوم أنت بإملاء رسالة طويلة على الهاتف. في سيناريو معتاد قد يقول آخذ الرسالة على الطرف الآخر من الخط "حسناً" بعد كل جملة يسمعها ويفهمها ويسجلها. إذا سمع آخذ الرسالة جملة مشوشة، فسوف يطلب منك تكرار الجملة المشوشة. يستخدم هذا البروتوكول لإملاء الرسالة الإشعار الإيجابي ("حسناً") والإشعار السلبي ("رجاءً كرّر تلك الجملة"). تسمح رسائل التحكم تلك للمُستقبِل بإعلام المُرسِل عما تم استلامه بشكلٍ صحيح، وما استُلم خطأ ومن ثم يحتاج إلى إعادة إرسال. في حالة شبكات الحاسب، يُطلق على بروتوكولات النقل الموثوق للبيانات التي تعتمد على مثل هذه الإعادة للإرسال بروتوكولات ARQ (طلب إعادة إرسال ذاتي Automatic Repeat Request).

يحتاج الأمر إلى ثلاث إمكانيات إضافية لتتمكن بروتوكولات ARQ من التعامل مع أخطاء البتات:

- اكتشاف الخطأ: نحتاج أولاً إلى آلية تُمكّن المُستقبِل من اكتشاف ما إذا كانت هناك أخطاء قد حدثت في البتات المُرسَلة. تذكّر من الجزء السابق أن بروتوكول UDP يستخدم حقل المجموع التدقيقي (checksum) لهذا الغرض. سنتناول في الفصل الخامس أساليب اكتشاف الأخطاء وتصحيحها بتفصيل أكثر؛ تلك الأساليب التي تسمح للمُستقبِل باكتشاف – وربما إصلاح – أخطاء البتات في الرزمة الواصلة. نحتاج الآن لنعرف فقط أن هذه الأساليب تتطلب إرسال بتات إضافية من المُرسِل إلى المُستقبِل (زيادة على بتات البيانات الأصلية المطلوب نقلها)، والتي تُوضع في حقل المجموع التدقيقي برزمة بيانات rdt2.0.
- التغذية المرتدة من المُستقبِل: لمّا كان المُرسِل والمُستقبِل يعملان عادةً على نظامين طرفيين مختلفين قد يفصل بينهما آلاف الأميال فإن الطريقة الوحيدة لكي يعرف المُرسِل وجهة نظر المُستقبِل عن العالم (في حالتنا هذه، ما إذا كانت الرزمة المُرسَلة قد تم استلامها بشكلٍ صحيح) هي أن يقوم المُستقبِل بتزويد المُرسِل بتغذية مرتدة واضحة. تُعتبر إشعارات الاستلام الإيجابية (ACK) والسلبية (NAK) في سيناريو إملاء الرسالة أمثلة لتلك التغذية المرتدة. بالمثل سيرسل بروتوكولنا rdt2.0 رزم ACK ورزم NAK كإشعارات استلام إيجابية وسلبية، على الترتيب، إلى المُرسِل. من حيث المبدأ تحتاج تلك الرزم إلى بت واحد لنقل تلك المعلومة؛ فعلى سبيل المثال يمكن أن يمثل 0 الإشعار السلبي (NAK) و1 الإشعار الإيجابي (ACK).
- إعادة الإرسال: سيقوم المُرسِل بإعادة إرسال الرزمة التي تصل إلى المُستقبِل وفيها أخطاء في البتات.

يبين الشكل 3-10 تمثيلاً لآلات الحالات المحدودة (FSM) لبروتوكول rdt2.0 لنقل البيانات والذي يستخدم أسلوباً لاكتشاف الأخطاء، وإشعارات استلام إيجابية (ACK)، وإشعارات استلام سلبية (NAK).



(a) بروتوكول rdt2.0: جانب الإرسال



(b) بروتوكول rdt2.0: جانب الاستقبال

الشكل 3-10 بروتوكول rdt2.0 لقناة تسبب أخطاءً في البتات.

يتضمن جانب المُرسِل في rdt2.0 حالتين. في الحالة المبيّنة أقصى اليسار يكون بروتوكول المُرسِل في انتظار عبور البيانات إليه من الطبقة الأعلى. على إثر الحدث rdt_send(data) سيقوم المُرسِل بتكوين رزمة sndpkt تتضمن البيانات المطلوب إرسالها، مع المجموع التدقيقي للرزمة (على سبيل المثال كما ذكرنا في حالة قطعة بيانات UDP بالجزء 3-3-2)، وبعد ذلك يُرسل الرزمة بواسطة عملية udt_send(sndpkt). أما في الحالة المبيّنة أقصى اليمين فينتظر بروتوكول المُرسِل وصول رزمة إشعار استلام (ACK أو NAK) من المُستقبِل. إذا تم استلام إشعار ACK

(يدل الرمز rdt_rcv(rcvpkt) && isACK(rcvpkt) في الشكل 3-10 على هذه الحالة) يدرك المُرسِل أن آخر رزمة أرسلها قد تم استلامها بشكلٍ صحيح بواسطة المُستقبِل، وهكذا يعود البروتوكول إلى حالة انتظار وصول بيانات من الطبقة الأعلى. إذا تم استلام إشعار NAK، يعيد البروتوكول إرسال الرزمة الأخيرة وينتظر وصول إشعار استلام (ACK أو NAK) من قبل المُستقبِل ردّاً على رزمة البيانات التي أعيد إرسالها. من المهم ملاحظة أنه عندما يكون المُرسِل في حالة انتظار وصول إشعار استلام (wait-for-ACK-or-NAK) لن يكون بوسعه تلقّي بيانات جديدة من الطبقة الأعلى لإرسالها، بمعنى أنه يتعذر حصول الحدث rdt_send() والذي سيحدث فقط بعد أن يتسلم المُرسِل إشعار الاستلام ACK ويغادر تلك الحالة. أي أن المُرسِل لن يبعث ببيانات جديدة إلا إذا تأكد من أن المُستقبِل قد تسلّم الرزمة الحالية بشكلٍ صحيح. بسبب هذا السلوك، يُطلق على البروتوكولات من نوع rdt2.0 بروتوكولات التوقّف والانتظار (stop-and-wait).

لا يزال لجانب المُستقبِل في rdt2.0 وضع واحد. عند وصول الرزمة يجيب المُستقبِل بإشعار استلام ACK أو NAK، حسب كون الرزمة المستلمة صحيحة أو فيها أخطاء. يدل الرمز rdt_rcv(rcvpkt) && corrupt(rcvpkt) في الشكل 3-10 على هذا الحدث في حالة استلام رزمة بها أخطاء.

قد يبدو البروتوكول rdt2.0 صحيحاً ويعمل على ما يرام، ولكنه لسوء الحظ يعاني من عيبٍ قاتل! بالتحديد لم نأخذ في الاعتبار حتى الآن احتمال حدوث أخطاء في رزمة إشعار الاستلام ACK أو NAK، (قبل متابعة القراءة، عليك أن تفكّر في طريقة لحل تلك المشكلة). لسوء الحظ فإن هفوتنا البسيطة ليست غير مؤذية كما قد يبدو للوهلة الأولى! كحد أدنى سنحتاج لإضافة حقل المجموع التدقيقي إلى رزم إشعار الاستلام ACK/NAK لاكتشاف مثل تلك الأخطاء. غير أن السؤال الأكثر صعوبة هو كيف يمكن أن يتعافى البروتوكول من الأخطاء في رزم إشعارات الاستلام؟ تكمن الصعوبة هنا في أنه في حالة فساد إشعار استلام نتيجة تلك الأخطاء لن يكون لدى المُرسِل أي طريقة لمعرفة ما إذا كان المستلم قد تلقّى قطعة البيانات الأخيرة المُرسَلة بشكلٍ صحيح.

خذ في الاعتبار البدائل الثلاثة التالية للتعامل مع فساد إشعارالاستلام ACK أو NAK نتيجةً لأخطاء البتات:

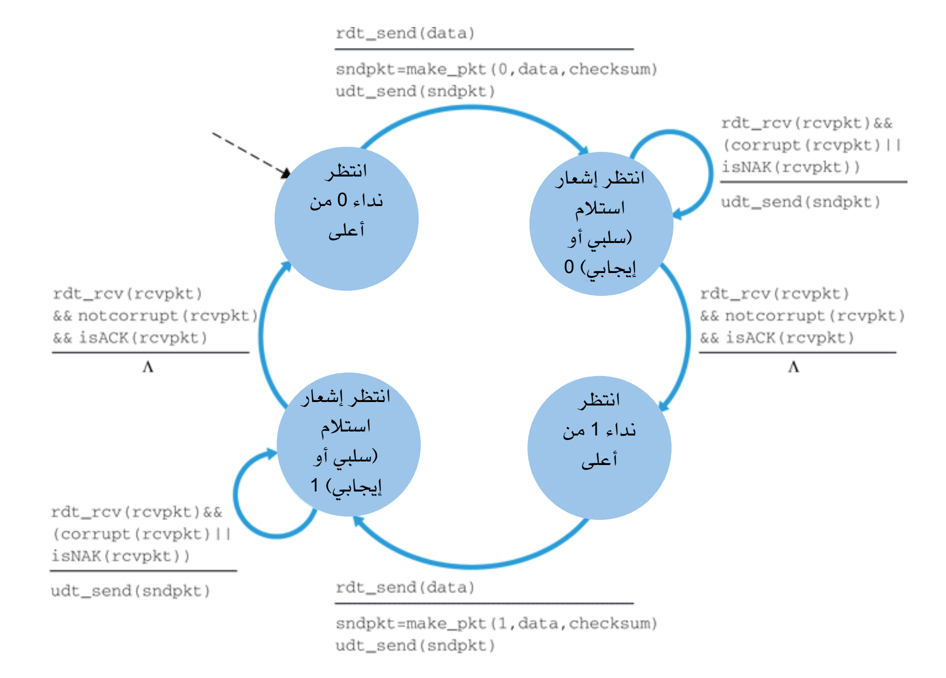
- البديل الأول: خذ في الاعتبار ماذا يمكن أن يحدث في المثال البشري الخاص بإملاء الرسائل. إذا لم يفهم المتكلم (مملي الرسالة) الإجابة "حسناً" أو "رجاءً كرّر" من آخذ الرسالة، فبوسع المتكلم أن يسأل: "ماذا قلت؟" (وبهذا نكون قد أضفنا نوعاً جديداً من الرزم من المُرسِل إلى المُستقبِل في بروتوكولنا). بعد ذلك يكرّر آخذ الرسالة إجابته. لكن ماذا لو أن "ماذا قلت؟" من المتكلم قد فسدت في الطريق هي الأخرى؟. في هذه الحالة لن يكون بوسع المُستقبِل تحديد ما إذا كانت الرزمة المشوهة التي وصلته جزءاً من الرسالة التي يجري إملاؤها أم طلباً لتكرار الإجابة الأخيرة. يُحتمل أن يرد هو الآخر عندئذ بـ "ماذا قلت؟" والتي بالطبع يمكن بدورها أن تفسد. واضح أننا بدأنا نسلك درباً وعِراً.
- يتلخص البديل الثاني في إضافة حقل من المجموع التدقيقي يكون كافياً ليس فقط لاكتشاف الأخطاء بواسطة المُرسِل، بل أيضاً للتعافي منها. هذا يحل المشكلة التي نحن بصددها لقناة اتصال يمكن أن تفسد عليها الرزم ولكنها لا تُفقد تماماً.
- كبديل ثالث يمكن للمُرسِل ببساطة إعادة إرسال رزمة البيانات الحالية عندما يتلقى إشعار استلام ACK أو NAK فاسد. على أية حال قد تؤدي هذه الطريقة إلى تكرار الرزم المُرسَلة عبر قناة المُرسِل-المُستقبِل. تكمن المشكلة الأساسية للرزم المكرّرة في أن المُستقبِل لا يعرف ما إذا كان آخر إشعار استلام ACK أو NAK أرسله قد وصل إلى المُرسِل بشكلٍ صحيح أم لا، ومن ثم فإنه لا يستطيع الجزم بكون الرزمة الواصلة تضم بيانات جديدة أم أنها مجرد إعادة إرسال!

هناك حل بسيط لهذه المشكلة الجديدة (وهو حل مستخدم تقريباً في كل بروتوكولات نقل البيانات الحالية، بما في ذلك TCP). يكمن الحل في إضافة حقل جديد إلى رزمة البيانات وجعل المُرسِل يرقّم رزم بياناته بوضع رقم تسلسلي في ذلك

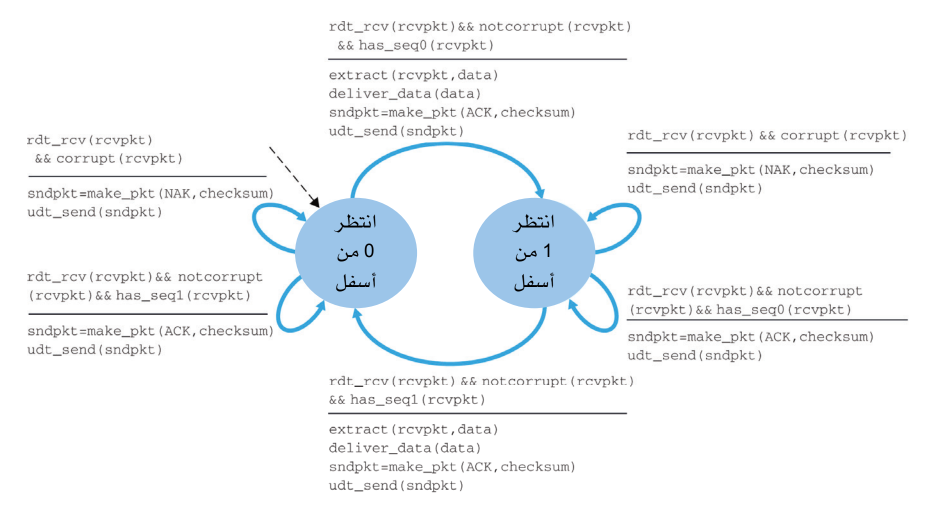
الحقل. عندئذ يحتاج المُستقبِل فقط لمراقبة هذا الرقم ليحدد ما إذا كانت الرزمة الواصلة جديدة أم إعادة إرسال. للحالة البسيطة لبروتوكول "توقّف وانتظر" الذي نحن بصدده، نحتاج لرقم تسلسلي حجمه بت واحدة فقط، حيث إن ذلك سيمكّن المُستقبِل من معرفة ما إذا كان المُرسِل يرسل للمرة الثانية الرزمة التي أرسلها سابقاً (الرقم التسلسلي للرزمة الجديدة هو نفسه للرزمة المستلمة مؤخراً) أو أنه يرسل رزمة جديدة (الرقم التسلسلي يتغيّر متحركاً "للأمام" حسب نظام حساب الباقي الثنائي (modulo-2 arithmetic). نظراً لأننا حالياً نفترض أن قناة الاتصال لا تفقد الرزم، فإن كلاً من إشعارات الاستلام نفسها ACK وNAK لا تحتاج لبيان الرقم المتسلسل للرزمة المتعلقة به. فالمُرسِل يعرف أن رزم ACK أو NAK (سواء كانت مشوشة أم لا) تم إرسالها من المُستقبِل كاستجابة لأحدث رزمة بيانات تم إرسالها.

يوضح الشكلان 3-11 و3-12 وصفاً لآلة FSM لبروتوكول rdt2.1 (إصدارنا المعدل لبروتوكول rdt2.0). لكلٍ من المُرسِل والمُستقبِل في هذا البروتوكول الآن ضعف عدد الأوضاع في الإصدار السابق. ذلك لأن حالة البروتوكول يجب أن تعكس الآن ما إذا كانت الرزمة التي ترسل حالياً (بواسطة المُرسِل) أو المتوقّع وصولها حالياً (عند المُستقبِل) لها الرقم التسلسلي 0 أو 1. لاحظ أن الإجراءات في تلك الأوضاع التي يتم فيها إرسال أو توقع رزمة رقم 0 هي صورة مطابقة لتلك التي يتم فيها إرسال أو توقع رزمة رقم 1 (الفرق الوحيد هو في التعامل مع الرقم المتسلسل).

يستخدم البروتوكول rdt2.1 كلاً من إشعارات الاستلام الإيجابية والسلبية من المُستقبِل إلى المُرسِل. عندما تصل رزمة برقم تسلسلي غير المتوقع، يُرسل المُستقبِل إشعار استلام إيجابي عن آخر رزمة تم استلامها. عندما تصل رزمة فيها أخطاء، يرسل المُستقبِل إشعار استلام سلبي. يمكننا الحصول على نفس تأثير الإشعار السلبي إذا أرسلنا بدلاً منه إشعار استلام إيجابي لآخر رزمة تم استلامها بشكلٍ صحيح. عندئذٍ سيعرف المُرسِل الذي يستلم إشعاري استلام إيجابيين لنفس الرزمة (أي يستلم إشعارات استلام مكرّرة) أن المُستقبِل لم يستلم بشكلٍ صحيح الرزمة التي أُرسلت بعد الرزمة التي وصل بخصوصها إشعاران إيجابيان.

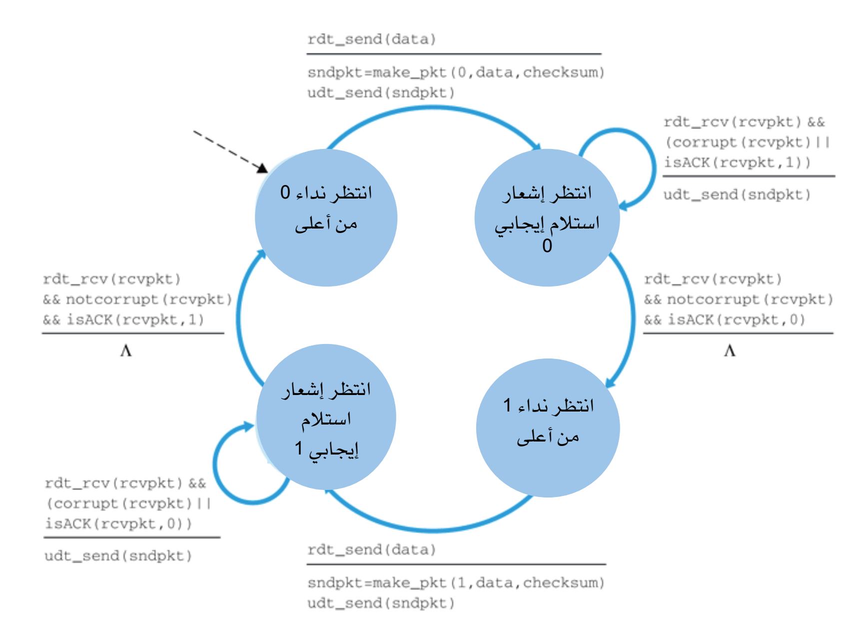


الشكل 3-11 مُرسِل بروتوكول rdt2.1.

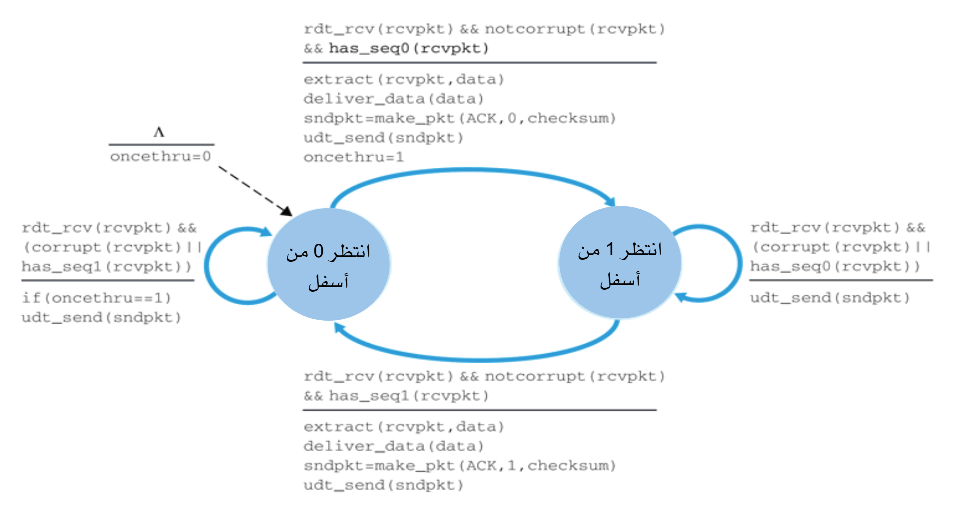


الشكل 3-12 مُستقبِل بروتوكول rdt2.1.

يبين الشكلان 3-13 و3-14 بروتوكولنا الجديد rdt2.2 للنقل الموثوق للبيانات على قناة اتصال بأخطاء في بتات القطعة وبدون استخدام إشعار الاستلام السلبي. من التغييرات الدقيقة بين rdt2.1 وrdt2.2 أن المُستقبِل عليه الآن أن يضمّن الرقم التسلسلي للرزمة في رسالة إشعار الاستلام ACK (وذلك بتضمين ACK,0 أو ACK,1 كمعامل في make_pkt() في آلة FSM الخاصة بالمُستقبِل)، كما أن على المُرسِل الآن فحص الرقم التسلسلي للرزمة الجاري إشعار الطرف الآخر باستلامها بواسطة رزمة ACK الواصلة (وذلك بتضمين الرقم 0 أو 1 كمعامل في isACK() في آلة FSM الخاصة بالمُرسِل).



الشكل 3-13 مُرسِل بروتوكول rdt2.2.



الشكل 3-14 مُستقبِل بروتوكول rdt2.2 .

النقل الموثوق للبيانات عبر قناة تُفقد فيها الرزم وتتعرض لأخطاء بتات: rdt3.0

افترض الآن أنه بالإضافة إلى فساد قطع البيانات بسبب أخطاء في البتات، يمكن أيضاً أن تفقد قناة الاتصال التحتية الرزم، وهذا شائع في شبكات الحاسب اليوم (بما في ذلك الإنترنت). هناك الآن اعتباران إضافيان يجب أن يعالجهما البروتوكول: كيف يمكن اكتشاف فقد رزمة وما الإجراء الذي ينبغي اتخاذه إزاء ذلك. إن استعمال المجموع التدقيقي، ورقم الرزمة المتسلسل، وإشعار استلام الرزمة، وإعادة الإرسال – وهي الأساليب التي طوّرناها في بروتوكول rdt2.2 – ستمكننا من التعامل مع الاعتبار الأخير، أما معالجة الاعتبار الأول فتتطلب إضافة آلية جديدة للبروتوكول.

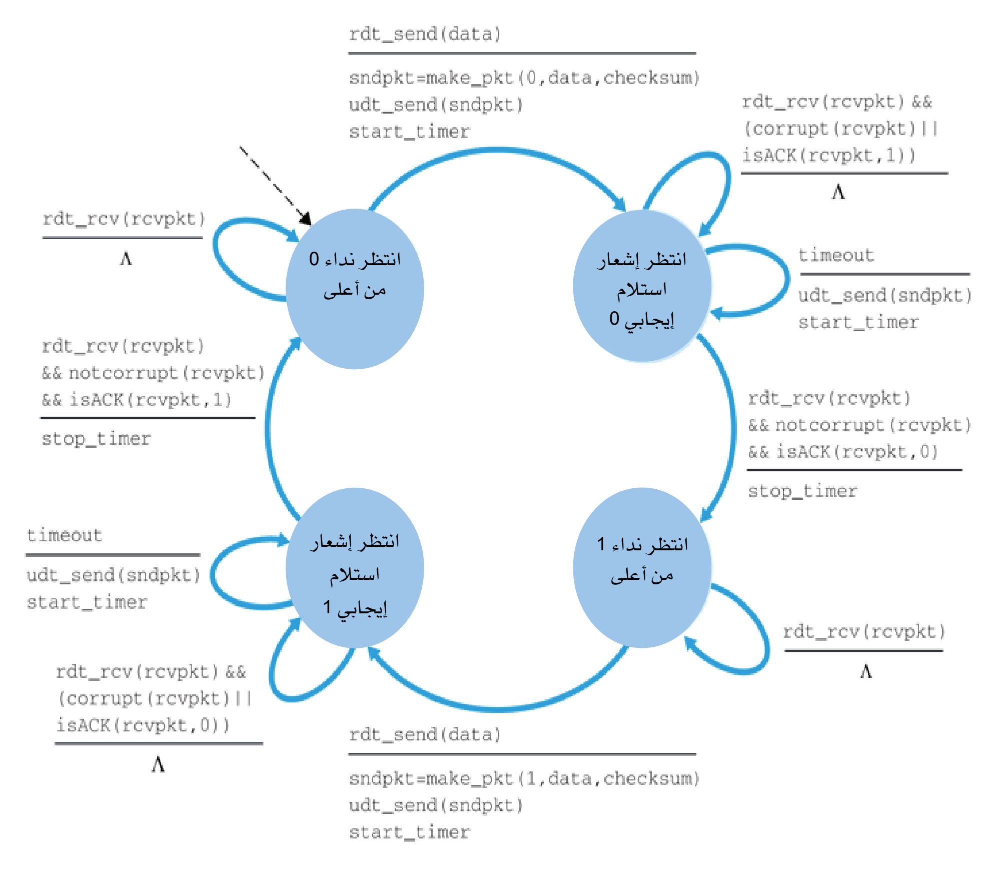
هناك العديد من الطرق للتعامل مع فقد الرزم (نستعرض المزيد منها في تمارين نهاية هذا الفصل). سنضع هنا عبء اكتشاف فقد الرزم والتعافي من ذلك على المُرسِل. افترض أن المُرسِل يرسل رزمة بيانات وأن تلك الرزمة أو إشعار الاستلام الخاص بها يفقد. في أي الحالتين لن تكون هناك إجابة قادمة من المُستقبِل إلى المُرسِل. إذا كان المُرسِل مستعداً للانتظار مدة طويلة تكفي للتأكد من أن رزمة قد فقدت، فبوسعه إعادة إرسال رزمة البيانات ببساطة بعد فترة الانتظار تلك. عليك أن تقنع نفسك بأن هذا البروتوكول يعمل في واقع الأمر!

ولكن إلى متى يجب على المُرسِل الانتظار ليتأكد من أن شيئاً ما قد فُقد؟ واضحٌ أنه على المُرسِل أن ينتظر على الأقل زمن التأخير المناظر لرحلة ذهاب وإياب الإشارة بين المُرسِل والمُستقبِل (والذي قد يتضمن زمن التخزين المؤقت في الموجّهات المتوسطة) بالإضافة إلى المدة اللازمة لمعالجة الرزمة لدى المُستقبِل. في العديد من الشبكات يصعب جداً تقدير هذا التأخير الأقصى لأسوأ الحالات، فضلاً عن تحديده بدقة. من ناحية أخرى وفي الوضع المثالي، على البروتوكول أن يتعافى من فقد الرزمة بأسرع ما يمكن، والانتظار لمدة تأخير مناظرة لأسوأ الحالات قد يعني انتظاراً طويلاً قبل أن تبدأ آليات التعافي من الخطأ في العمل. إن الطريقة المتبعة عملياً هي أن يختار المُرسِل بحذر قيمةً لوقت التأخير يكون بعدها فقد الرزمة محتمل الحدوث، حتى إن لم يكن الفقد مؤكداً. فإذا لم يصل إلى المُرسِل إشعار استلام ACK في غضون ذلك الوقت فإنه يعيد إرسال الرزمة على أي حال. لاحظ أنه إذا واجهت رزمة تأخيراً كبيراً جداً، فإن المُرسِل قد يعيد إرسال رزمة رغم عدم فقدها أو فقد إشعار استلامها، مما قد يؤدي إلى احتمال إرسال رزم مكرّرة من المُرسِل إلى المُستقبِل عبر قناة الاتصال. لحسن الحظ تتوافر لبروتوكول rdt2.2 إمكانية التعامل مع حالة الرزم المكرّرة (عن طريق الأرقام التسلسلية للرزم).

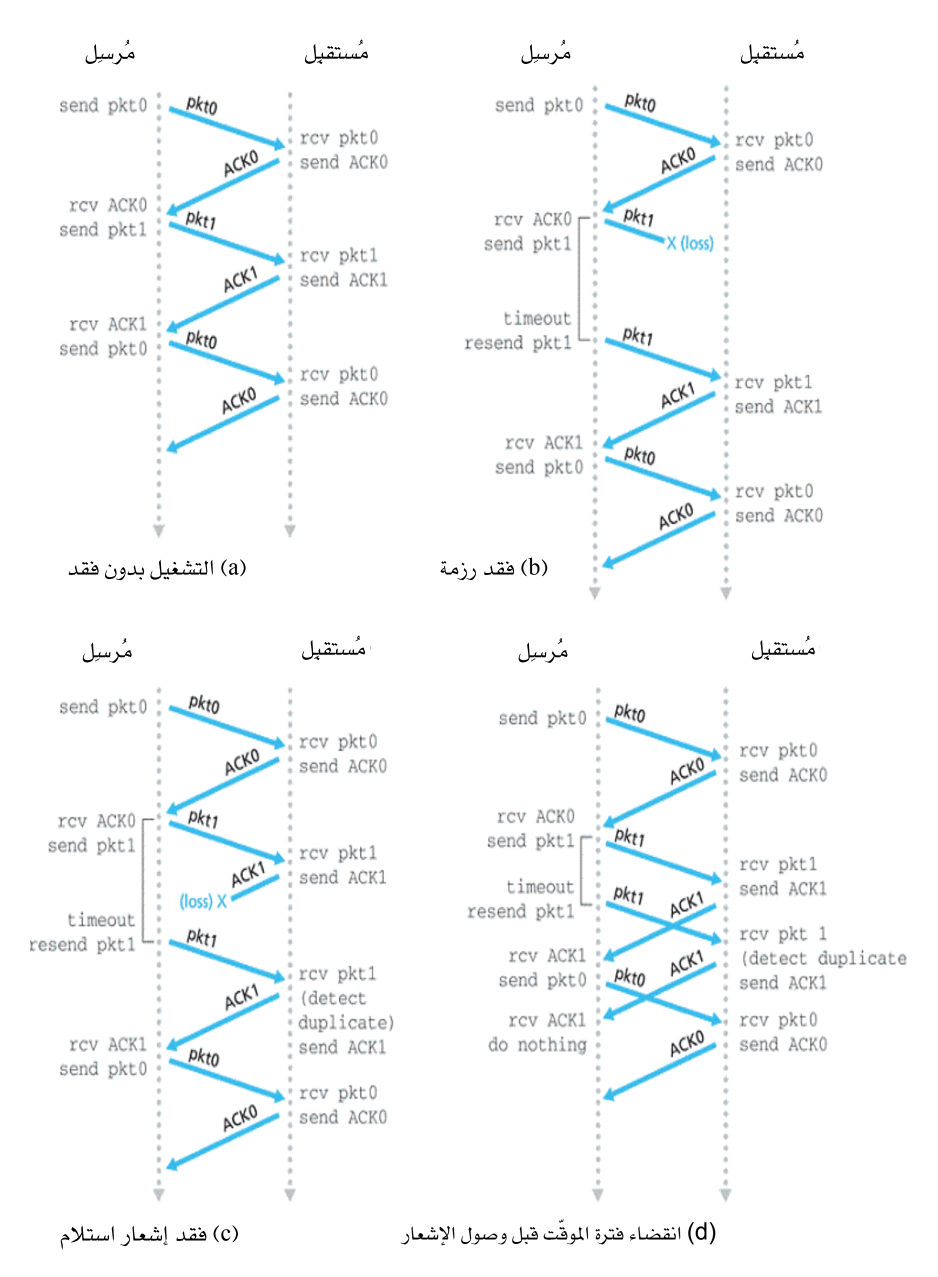
من وجهة نظر المُرسِل تعتبر إعادة الإرسال حلاً ناجعاً، فالمُرسِل لا يعرف ما إذا كانت رزمة بيانات قد فُقدت، أو أن إشعار الاستلام الخاص بها قد فُقد، أو أن الرزمة أو إشعار الاستلام ببساطة قد تأخرا في الطريق أكثر من اللازم. في كل تلك الحالات الإجراء المتخذ واحد: "أعِد الإرسال". يحتاج تنفيذ آلية إعادة الإرسال إلى موقّت بعَدٍّ تنازليٍّ يقوم بمقاطعة (interrupt) المُرسِل بعد انقضاء فترة زمنية معينة يتم برمجتها مسبقاً. مما تقدم يتعين على المُرسِل أن يكون قادراً على: (1) بدأ الموقّت عند إرسال كل رزمة (سواء كانت رزمة جديدة أو رزمة معاداً إرسالها)، (2) الاستجابة لمقاطعة الموقّت (باتخاذ التدابير الملائمة)، و(3) إيقاف الموقّت.

يبين الشكل 3-15 آلة FSM للمُرسِل في بروتوكول rdt3.0 للنقل الموثوق للبيانات على قناة اتصال يمكن أن تُفسد أو تفقد الرزم. في تمارين نهاية الفصل، سيُطلب منك تصميم آلة FSM للمُستقبِل في بروتوكول rdt3.0. يوضح الشكل 3-16 كيف يعمل البروتوكول بدون رزم مفقودة أو متأخرة وكيف يعالج فقد رزم البيانات. في الشكل 3-16 يتقدّم الوقت للأمام من أعلى الشكل إلى أسفله. لاحظ أن وقت استلام رزمة يأتي بالضرورة بعد وقت إرسال الرزمة نتيجة لتأخيرات

الإرسال والانتقال. في الأشكال 3-16 (b)-(d) تشير الأقواس الجانبية إلى الأوقات التي يضبط عندها الموقّت والأوقات التي تنتهي فيها مدته لاحقاً. يتم استكشاف العديد من السمات الدقيقة لهذا البروتوكول في التمارين في نهاية هذا الفصل. نظراً لأن الأرقام المتسلسلة للرزم تتبادل ما بين 0 و1 ، فإن بروتوكول rdt3.0 يعرف أحيانا ببروتوكول البت المتناوبة (alternating bit protocol).



الشكل 3-15 مُرسِل بروتوكول rdt 3.0.



الشكل 3-16 طريقة عمل بروتوكول rdt 3.0 (بروتوكول البت المتناوبة).

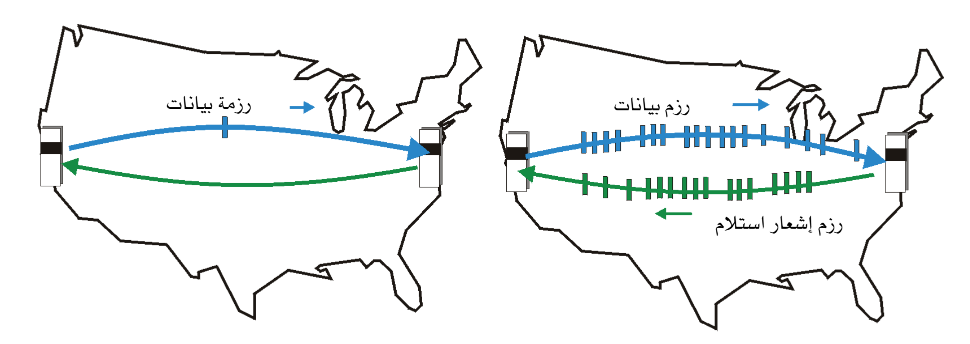
لقد جمّعنا الآن العناصر الرئيسة لبروتوكول نقل البيانات. يلعب كلٌّ من المجموع التدقيقي، والأرقام التسلسلية، والموقّتات، وإشعارات الاستلام الإيجابية والسلبية دوراً حاسماً وضرورياً في عمل البروتوكول. أخيراً أصبح لدينا الآن بروتوكولاً يصلح لنقل البيانات بشكلٍ موثوق!

3-4-2 بروتوكولات النقل الموثوق للبيانات بأسلوب خط الأنابيب

رغم أن بروتوكول rdt3.0 صحيحٌ إجرائياً، فإنه من غير المحتمل أن يكون أحدٌ سعيداً بأدائه، خصوصاً في شبكات اليوم السريعة. فأساس مشكلة أداء بروتوكول rdt3.0 أنه نظام توقّف وانتظار.

لإدراك تأثير التوقّف والانتظار على أداء البروتوكول، تصور حالة مثالية: مضيفين أحدهما يقع على الساحل الغربي للولايات المتحدة والآخر يقع على الساحل الشرقي، كما هو مبين في الشكل 3-17. يبلغ تأخير انتقال الضوء في رحلة الذهاب والإياب (RTT) بين هذين النظامين الطرفيين حوالي 30 ميللي ثانية. افترض أنهما موصلان عن طريق قناة اتصال لها معدل إرسال للبيانات (R) قدره 1 جيجابت/ثانية (= 10^9 بت/ثانية)، وأن حجم كل رزمة $L = 1000$ بايت بما في ذلك حقول البيانات والترويسة، وبالتالي يكون الوقت اللازم لإرسال تلك الرزمة على الوصلة هو:

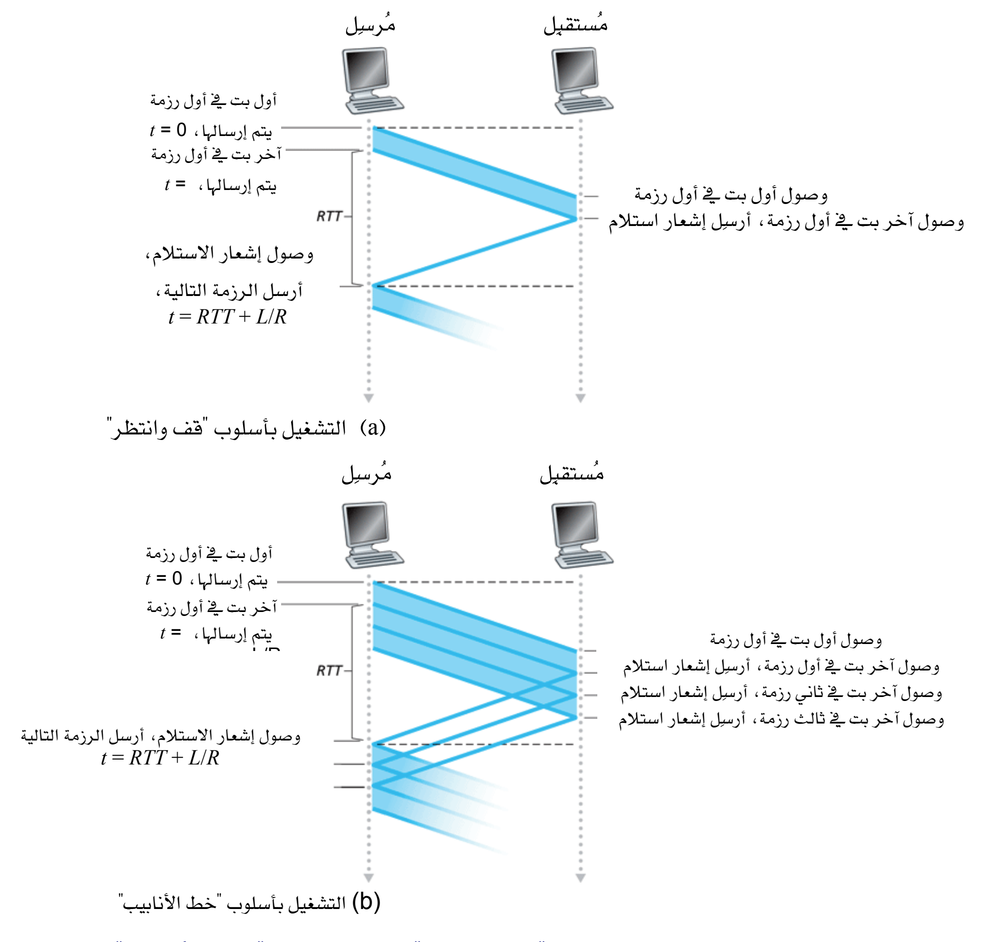
$$d_{trans} = \frac{L}{R} = \frac{8 \times 1000\ bits/packet}{10^9 bits/\sec} = 8\ \mu\sec$$



الشكل 3-17 بروتوكول "قف وانتظر" في مقابل بروتوكول "خط الأنابيب".

يبين الشكل 3-18 (a) أنه باستخدام بروتوكول من نوع "توقّف وانتظر"، إذا بدأ المُرسِل بإرسال رزمة عند زمن t = 0، فإنه عند t =L/R = 8 ميكروثانية، تدخل البت الأخيرة من الرزمة القناة في جانب المُرسِل. تقطع الرزمة رحلتها بعرض الولايات المتحدة في 15 ميللي ثانية، ومن ثم تصل البت الأخيرة من الرزمة إلى المُستقبِل عند $t = L/R + RTT/2 = 15.008$ ميللي ثانية. افترض للتبسيط أن رزم إشعار الاستلام صغيرة جداً (بحيث يمكننا إهمال وقت إرسالها) وأنه بوسع المُستقبِل إرسال إشعار استلام بمجرد وصول البت الأخيرة من رزمة البيانات إليه، وبذلك يصل إشعار الاستلام إلى المُرسِل عند $t = L/R + RTT = 30.008$ ميللي ثانية. في تلك اللحظة يمكن للمُرسِل البدء في بث الرزمة التالية. وهكذا فخلال 30.008 ميللي ثانية، تمكّن المُرسِل من بث البيانات لمدة 0.008 ميللي ثانية فقط. إذا عرّفنا استغلال المُرسِل (أو القناة) (utilization) على أنه كسر الوقت الذي يكون فيه المُرسِل مشغولاً ببث البيانات بالفعل إلى القناة، فإنه بدراسة الشكل 3-18 (a) يتضح أن نظام التوقّف والانتظار يعطي استغلالاً ضئيلاً للمُرسِل قيمته:

$$U_{sender} = \frac{L/R}{RTT + L/R} = \frac{0.008}{30.008} = 0.00027$$



الشكل 3-18 الإرسال ببروتوكول "قف وانتظر" و بروتوكول "خط الأنابيب".

أي أن المُرسِل يكون مشغولاً بالإرسال فقط بنسبة 2.7 جزء من كل عشرة آلاف جزء من الوقت! من منظور آخر، يكون المُرسِل قادراً على إرسال 1000 بايت فقط خلال 30.008 ميللي ثانية، أي بطاقة إنتاجية فعلية مقدارها 267 كيلو بت/ثانية فقط رغم توفر استخدام قناة سعة إرسالها 1 جيجابت/ثانية. تخيل موقف مدير الشبكة المستاء الذي دفع ثروة طائلة لتوفير وصلة بسعة إرسال 1 جيجابت/ثانية ولكنه لا يستطيع سوى تأمين طاقة إنتاجية لا تعدو 267 كيلوبت/ثانية فقط. هذا مثال رسومي يوضح كيف يمكن لبروتوكولات

الشبكة أن تُحِدّ من استغلال الإمكانيات التي توفرها البنية التحتية للشبكة. تذكّر أننا أهملنا أيضاً أوقات المعالجة لبروتوكولات الطبقات الأدنى في المُرسِل والمُستقبِل، بالإضافة إلى أوقات المعالجة وتأخيرات الانتظار في الصف التي يمكن أن تحدث في أي من الموجّهات المتوسّطة بين المُرسِل والمُستقبِل. لاحظ أن أخذ تلك التأثيرات بعين الاعتبار سيزيد من التأخير أكثر، وبالتالي سيزيد الأداء السيّئ سوءاً.

إن حل مشكلة الأداء هذه بسيط من حيث المبدأ: بدلاً من العمل بطريقة توقّف وانتظر، يُسمح للمُرسِل بإرسال عدة رزم بدون انتظار لإشعارات الاستلام، كما يبيّن الشكل 3-17 (b). يوضح الشكل 3-18 (b) أنه إذا ما سُمح للمُرسِل بإرسال ثلاثة رزم قبل الحاجة لانتظار إشعارات الاستلام، فإن نسبة الاستغلال عند المُرسِل تزداد إلى ثلاثة أضعاف تقريباً. لمّا كانت رزم البيانات العديدة المرصوصة على الوصلة في نفس الوقت في طريقها من المُرسِل إلى المُستقبِل يمكن تصورها كما لو كانت تملأ خط أنابيب، تعرف هذه الطريقة بأسلوب خط الأنابيب (pipeline)، ولهذا الأسلوب الانعكاسات التالية على بروتوكولات النقل الموثوق للبيانات:

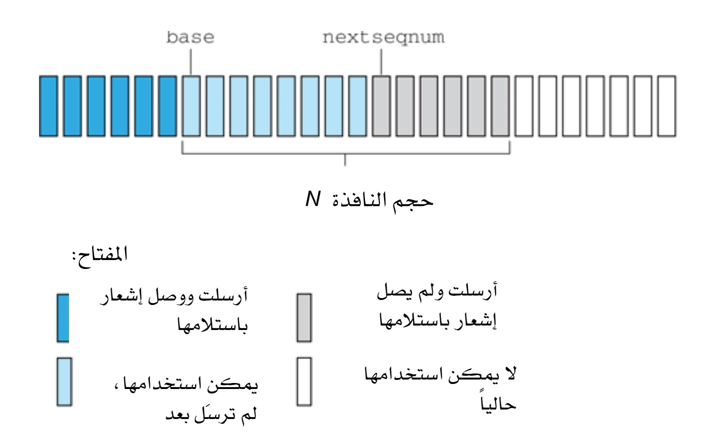
- ينبغي زيادة مدى الأرقام التسلسلية للرزم، نظراً لأنه يجب أن يكون لكل الرزم التي ما زالت في طريقها إلى المُستقبل (باستثناء الرزم التي يعاد إرسالها) رقم تسلسلي فريد، كما يمكن أن تكون هناك عدة رزم لم يتم الإشعار عن استلامها.
- قد يحتاج الأمر إلى التخزين المؤقت لأكثر من رزمة لدى كلٍّ من جانبي المُرسِل والمُستقبِل من البروتوكول، لذا يتعين على المُرسِل كحد أدنى توفير حيّز تخزين مؤقت للرزم التي تم إرسالها ولم تصل إشعارات استلامها بعد. أحياناً يحتاج المُستقبِل أيضاً لتخزين رزم تم استلامها بشكلٍ صحيح، كما سنرى لاحقاً.
- يعتمد مدى الرقم التسلسلي المطلوب ومتطلبات التخزين المؤقت اللازمة على طريقة تعامل بروتوكول نقل البيانات مع الرزم المفقودة

والفاسدة والمتأخرة تأخيراً غير عادي. هناك طريقتان أساسيتان للتعامل مع الخطأ في بروتوكول المعالجة بأسلوب خط الأنابيب: طريقة "ارجع N للوراء" ((Go-Back-N (GBN)) وطريقة "الإعادة الانتقائية" ((Selective Repeat (SR)).

3-4-3 بروتوكول "ارجع N للوراء" (GBN)

في بروتوكول "ارجع N للوراء" يُسمح للمُرسِل بإرسال عدة رزم (عند توفرها) بدون انتظار وصول إشعارات الاستلام، لكن الحد الأقصى المسموح به هو عدد N من تلك الرزم، سنتناول هنا بروتوكول GBN بشيء من التفصيل، لكن قبل مواصلة القراءة ننصحك بتشغيل برنامج جافا GBN applet الموجود في موقع الويب المصاحب لهذا الكتاب (وهو برنامج توضيحي رائع!).

يبين الشكل 3-19 مدى الأرقام التسلسلية للرزم من منظور المُرسِل في بروتوكول GBN. إذا عرّفنا base بأنه الرقم التسلسلي لأقدم رزمة أرسلت ولم يصل بعد إشعار باستلامها، و nextseqnum بأنه أصغر رقم تسلسلي لم يستخدم بعد (أي الرقم التسلسلي للرزمة التي سترسل في المرة القادمة)، عندئذٍ يمكن تحديد أربع فترات في مدى الأرقام التسلسلية: الأرقام في الفترة [0, base-1] وتناظر الرزم التي أرسلت ووصلت إشعارات باستلامها. والفترة [base, nextsegnum-1] وتناظر الرزم التي أرسلت ولكن لم تصل إشعارات باستلامها بعد، والفترة [nextsegnum, base +N-1] وتناظر الرزم التي يمكن أن ترسل فور وصول بيانات من الطبقة الأعلى، وأخيراً الفترة التي فيها الرقم التسلسلي ≤ base+N والتي لن يتسنى استخدام أرقامها للإرسال قبل وصول إشعار استلام لرزمة مُرسَلة وموجودة حالياً في خط الأنابيب (بالتحديد رزمة تحمل الرقم التسلسلي base).



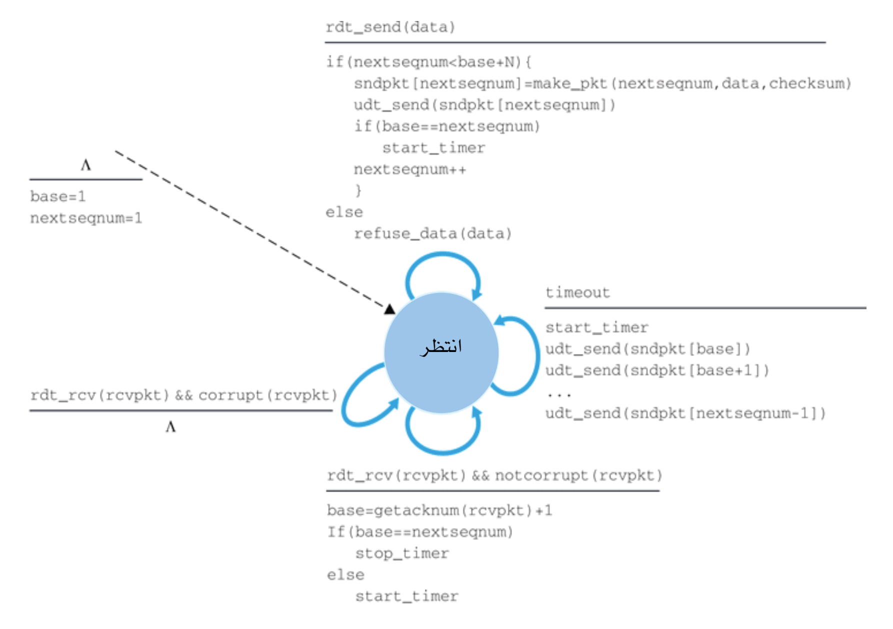
الشكل 3-19 الأرقام التسلسلية من منظور المُرسِل في بروتوكول العودة N للوراء (GBN).

من خلال الشكل 3-19 يمكن اعتبار مدى الأرقام التسلسلية المسموح به لإرسال الرزم مع عدم وصول إشعارات استلامها على أنه نافذة مقاسها N. وأثناء تشغيل البروتوكول تنزلق تلك النافذة للأمام على مدى الأرقام التسلسلية. لهذا السبب غالباً ما يطلق على N حجم النافذة وعلى البروتوكول نفسه بروتوكول النافذة المنزلقة (sliding window protocol). قد تتساءل لماذا نُحد من عدد الرزم المنتظرة بدون إشعار استلام بالقيمة N في المقام الأول؟ لماذا لا نسمح لعدد غير محدود من تلك الرزم؟ سنرى في الجزء 3-5 أن ضبط التدفق يمثل أحد الأسباب لوضع ذلك القيد على المُرسِل. سنرى سبباً آخر لذلك في الجزء 3-7 عندما ندرس السيطرة على الازدحام في بروتوكول TCP.

عملياً يُوضع الرقم التسلسلي للرزمة في حقل طوله ثابت في ترويسة الرزمة. إذا كانت k هي عدد البتات في حقل الرقم التسلسلي للرزمة فإن مدى الأرقام التسلسلية يكون $[0, 2^k-1]$. في وجود مدى محدود من تلك الأرقام التسلسلية، يتعين أن تُجرى كل الحسابات المتعلقة بتلك الأرقام التسلسلية على أساس modulo-2^k (بمعنى أن فضاء الأرقام التسلسلية يمكن تصوره كحلقة مقاسها 2^k، حيث يأتي الرقم 0 بعد الرقم 2^k-1). تذكّر أنه في البروتوكول rdt3.0 كانت الأرقام التسلسلية لها حقل يتكون من بت واحدة ومدى [0،1]. من خلال عدة تمارين في نهاية هذا

الفصل ستكتشف النتائج المترتبة على المدى المحدود للأرقام التسلسلية. سنرى في الجزء 3-5 أنه في بروتوكول TCP يتألف حقل الرقم التسلسلي من 32 بتاً، حيث تعِدّ الأرقام التسلسلية في هذا البروتوكول البايتات المُرسَلة بدلاً من الرزم.

يصور الشكلان 3-20 و 3-21 وصفاً لآلة FSM موسعة لكلٍ من جانبي المُرسِل والمُستقبِل لبروتوكول مبني على إشعارات الاستلام الإيجابية ACK وبدون إشعارات استلام سلبية NAK ويستخدم طريقة "ارجع N للوراء" (GBN). أطلقنا وصف "موسّعة" على آلة FSM تلك لأننا أضفنا متغيرات (تشبه المتغيرات المستخدمة في لغة البرمجة) لكلٍ من base و nextseqnum، بالإضافة إلى عمليات وإجراءات شرطية تتعلق بهذين المتغيرين. لاحظ أن مواصفات آلة FSM الموسعة الآن أخذت تشبه بعض الشيء مواصفات لغة البرمجة. يعطي [Bochman 1984] استعراضاً ممتازاً للامتدادات الإضافية لأساليب FSM وغيرها من الأساليب المبنية على لغات البرمجة والمستخدمة لتوصيف البروتوكولات.



الشكل 3-20 وصف لآلة الأوضاع المحدودة (FSM) الموسّعة للمُرسِل في بروتوكول العودة N للوراء (GBN).

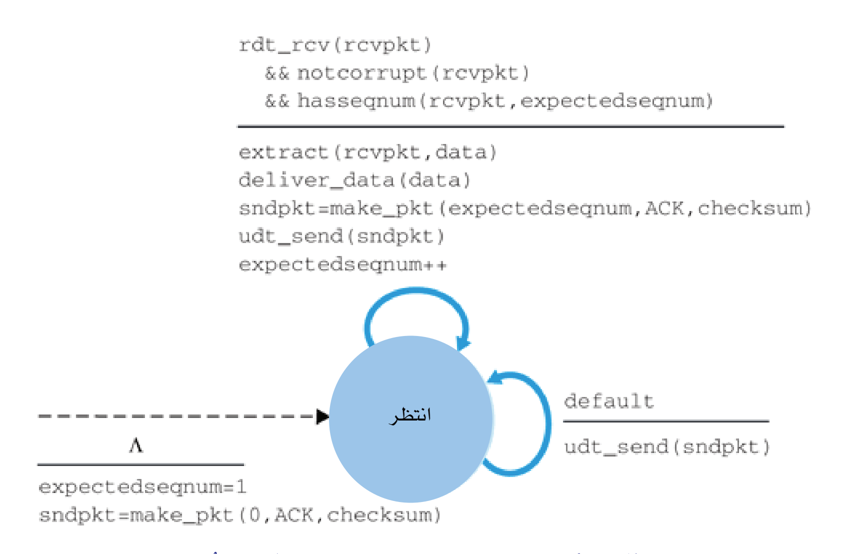
ينبغي على مُرسِل بروتوكول GBN الاستجابة لثلاثة أنواع من الأحداث:

- استدعاء من أعلى: عندما يحدث نداء لـ rdt_send() من الطبقة الأعلى، يتأكد المُرسِل أولاً مما إذا كانت النافذة ممتلئة، بمعنى أن هناك N رزمة مُرسَلة لم تصل إشعارات استلامها. فإذا كانت النافذة ليست ممتلئة، يقوم المُرسِل بتكوين رزمة وإرسالها، ثم بتحديث المتغيرات بالشكل الملائم، وإذا كانت النافذة ممتلئة، يُعيد المُرسِل البيانات ببساطة إلى الطبقة الأعلى، في إشارة ضمنية لأن النافذة ممتلئة. يُفترض أن تعاود الطبقة الأعلى محاولة الإرسال من جديد لاحقاً. في تطبيق حقيقي للبروتوكول قد يقوم المُرسِل على الأرجح بتخزين تلك البيانات عنده بشكلٍ مؤقت (ولكنه لا يرسلها على الفور)، أو تكون لديه آلية للتزامن (على سبيل المثال سيمافور (semaphore) أو علَم (flag) تسمح للطبقة الأعلى باستدعاء rdt_send() فقط عندما تكون النافذة غير ممتلئة.
- تلقّي إشعار استلام: في بروتوكول من نوع GBN سيُأخذ إشعار الاستلام لرزمة تحمل الرقم التسلسلي n على أنه إشعار تراكمي بأن كل الرزم التي تحمل الأرقام التسلسلية حتى (وبما في ذلك) الرقم n قد تم استلامها بشكلٍ صحيح في المُستقبِل. لنا عودة إلى هذا الوضع بعد قليل عندما نتناول جانب المُستقبِل من بروتوكول GBN.
- انقضاء فترة الموقّت: تذكّر أن اسم البروتوكول "ارجع N للوراء" مشتق من سلوك المُرسِل إزاء الرزم المفقودة أو التي تأخرت كثيراً. كما في نظام التوقّف والانتظار، يُستخدم هنا موقّت للتعافي من حالات فقد رزم البيانات أو رزم إشعارات الاستلام. إذا انقضت فترة الموقّت يقوم المُرسِل بإعادة إرسال كل الرزم التي أُرسلت سابقاً ولم تصل إشعارات باستلامها بعد. يستخدم مُرسِلنا في الشكل 3-20 موقّتاً واحداً فقط، والذي يمكن اعتباره موقّتاً لأقدم رزمة تم إرسالها ولكن لم يصل بعد إشعار استلامها. إذا لم تكن هناك أي رزم متبقية بدون وصول إشعار باستلامها، يتم إيقاف الموقّت قبل انقضاء فترته.

إنّ أعمال المُستقبِل في بروتوكول GBN هي أيضاً بسيطة. إذا وصلت رزمة برقم تسلسلي n وتم استلامها صحيحة وبترتيب سليم (بمعنى أن آخر بيانات تم

رفعها للطبقة الأعلى تم استخراجها من رزمة لها الرقم التسلسلي $n - 1$)، يرسل المُستقبِل إشعار استلام بخصوص الرزمة n ويقوم بتوصيل جزء البيانات من الرزمة إلى الطبقة الأعلى. في كل الحالات الأخرى يهمل المُستقبِل الرزمة الحالية ويعيد إرسال إشعار استلام لأحدث رزمة تم استلامها بالترتيب السليم. لاحظ أنه نظراً لأن الرزم يتم توصيلها الواحدة تلو الأخرى إلى الطبقة الأعلى، فإنه في حالة استلام الرزمة k بواسطة المُستقبِل وتوصيلها للطبقة الأعلى، فإن كل الرزم برقم تسلسلي أقل من k تكون قد سُلّمت أيضاً. وهكذا فإن الاستعمال التراكمي لإشعارات الاستلام يعتبر اختياراً طبيعياً في بروتوكول GBN.

في بروتوكول GBN يُهمل المُستقبِل الرزم التي تصل في غير ترتيبها السليم. ورغم أن إهمال رزم تصل صحيحة ولكن بترتيب خطأ قد يبدو تصرفاً غير حكيم وينطوي على شيء من التبذير، فإن هناك بعض التبريرات لذلك. تذكّر أنه على المُستقبِل توصيل البيانات للطبقة الأعلى بالترتيب. افترض الآن أن الحزمة n متوقّعة ولكن الرزمة $n + 1$ هي التي وصلت. نظراً لأن البيانات ينبغي توصيلها بالترتيب، فإنه يجب على المُستقبِل هنا أن يخزن الرزمة $n + 1$ لديه مؤقتاً على أن يسلّم تلك الرزمة إلى الطبقة الأعلى لاحقاً بعد أن يكون قد استلم الرزمة n وسلّمها للطبقة الأعلى. ومع ذلك فإذا فقدت الرزمة n، فإن كلاً من تلك الرزمة والرزمة التالية لها ($n + 1$) سيعاد إرسالهما في النهاية كنتيجة لقواعد بروتوكول GBN التي تحكم إعادة الإرسال. وعليه يمكن للمستلم ببساطة أن يهمل الرزمة $n + 1$. تكمن ميزة هذه الطريقة في تبسيط آليات التخزين المؤقت لدى المُستقبِل، حيث لا يحتاج المُستقبِل للتخزين المؤقت للرزم التي لا تأتي في ترتيبها السليم. وهكذا فبينما يتعين على المُرسِل الإبقاء على الحدود العليا والدنيا لنافذته وموضع nextseqnum ضمن حدود تلك النافذة، فإن المعلومة الوحيدة التي يجب على المُستقبِل الاحتفاظ بها هي الرقم التسلسلي للرزمة التالية حسب الترتيب السليم، تحفظ هذه القيمة في المتغير expectedseqnum، والموضحة في آلة FSM للمُستقبِل في الشكل 3-21. بالطبع فإن من عيوب إهمال رزمة صحيحة أن إعادة إرسال تلك الرزمة لاحقاً قد يتعرض للفقد أو لفساد البيانات، مما يتطلب بدوره المزيد من إعادة الإرسال.

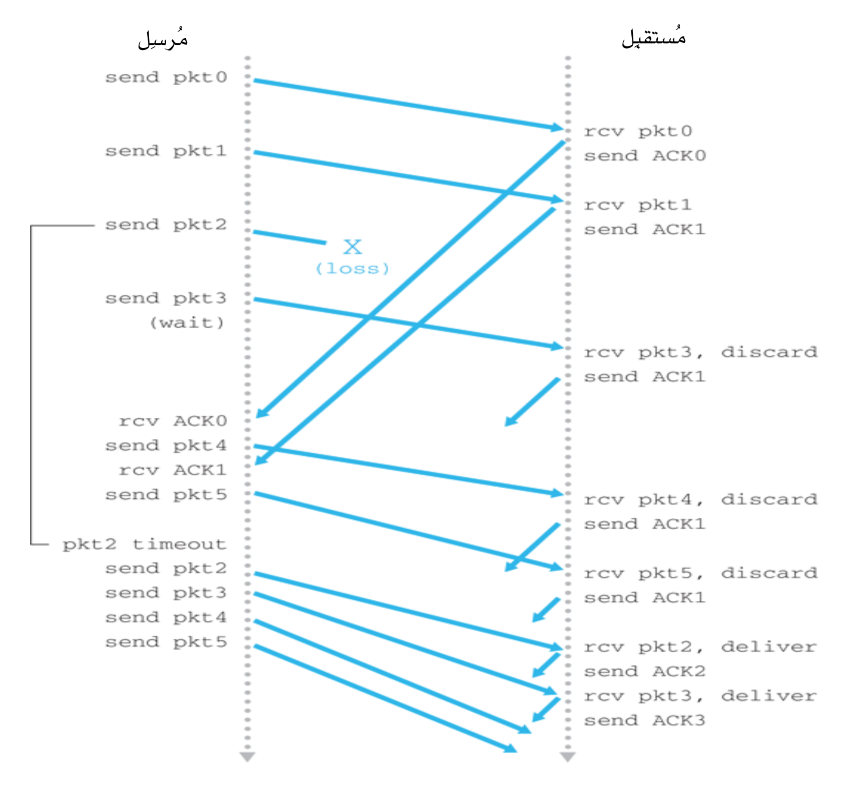


الشكل 3-21 وصف لآلة الأوضاع المحدودة (FSM) الموسّعة للمُستقبِل في بروتوكول العودة N للوراء (GBN).

يبين الشكل 3-22 عمل بروتوكول GBN عند استخدام نافذة حجمها أربع رزم. بسبب التقيد بسعة النافذة يرسل المُرسِل الرزم التي تحمل الأرقام التسلسلية من 0 إلى 3، ولكن عليه بعد ذلك أن ينتظر لحين وصول إشعارات استلام لواحد أو أكثر من تلك الرزم قبل المُضي قدُماً في إرسال المزيد من الرزم. وبوصول إشعارات الاستلام الواحد تلو الآخر (مثلاً ACK0 ثم ACK1)، تنزلق النافذة للأمام ويقوم المُرسِل ببث رزمٍ جديدة (رزمة 4 ثم رزمة 5 على التوالي). على جانب المُستقبِل تفقد الرزمة 2 ومن ثم تصل الرزم 3 و4 و5 بغير الترتيب السليم وبالتالي يتم إهمالها.

قبل أن ننهي مناقشتنا لبروتوكول GBN يجدر بنا ملاحظة أن تطبيقه في رصة البروتوكولات يتطلب في الغالب هيكلاً يشبه هيكل آلة FSM الموسّعة في الشكل 3-20، وغالباً ما سيأخذ هذا التطبيق أيضاً شكل إجراءات مختلفة لتنفيذ الأعمال المطلوب تنفيذها كاستجابة للأحداث المختلفة التي يمكن أن تقع. في مثل هذه البرمجة المبنية على الحدث (event-driven programming) تم استدعاء (تفعيل) الإجراءات المختلفة بواسطة إجراءات أخرى في رصّة البروتوكولات، أو كنتيجة

لحدوث مقاطعة (interrupt). بالنسبة للمُرسِل تتضمن هذه الأحداث ما يلي: (1) نداء من الكيان في الطبقة الأعلى لتشغيل rdt_send()، (2) مقاطعة من موقّت، (3) نداء من الطبقة الأسفل لتشغيل rdt_rcv() عند وصول رزمة. تتيح لك تمارين البرمجة في نهاية هذا الفصل الفرصة للتطبيق الحقيقي لتلك البرامج في محاكاة لمواقف واقعية لشبكات الحاسب.



الشكل 3-22 العمل ببروتوكول العودة N للوراء GBN.

نلاحظ هنا أن بروتوكول GBN يتضمن تقريباً كل الأساليب التي سوف نستعرضها عند دراستنا لمكوّنات بروتوكول TCP للنقل الموثوق للبيانات في الجزء 3-5. تتضمّن تلك الأساليب استعمال الأرقام التسلسلية، وإشعارات الاستلام

التراكمية، وحقل المجموع التدقيقي لاكتشاف أخطاء البيانات، وإعادة الإرسال بناءً على انقضاء فترة الموقّت.

3-4-4 بروتوكول "الإعادة الانتقائية" (SR)

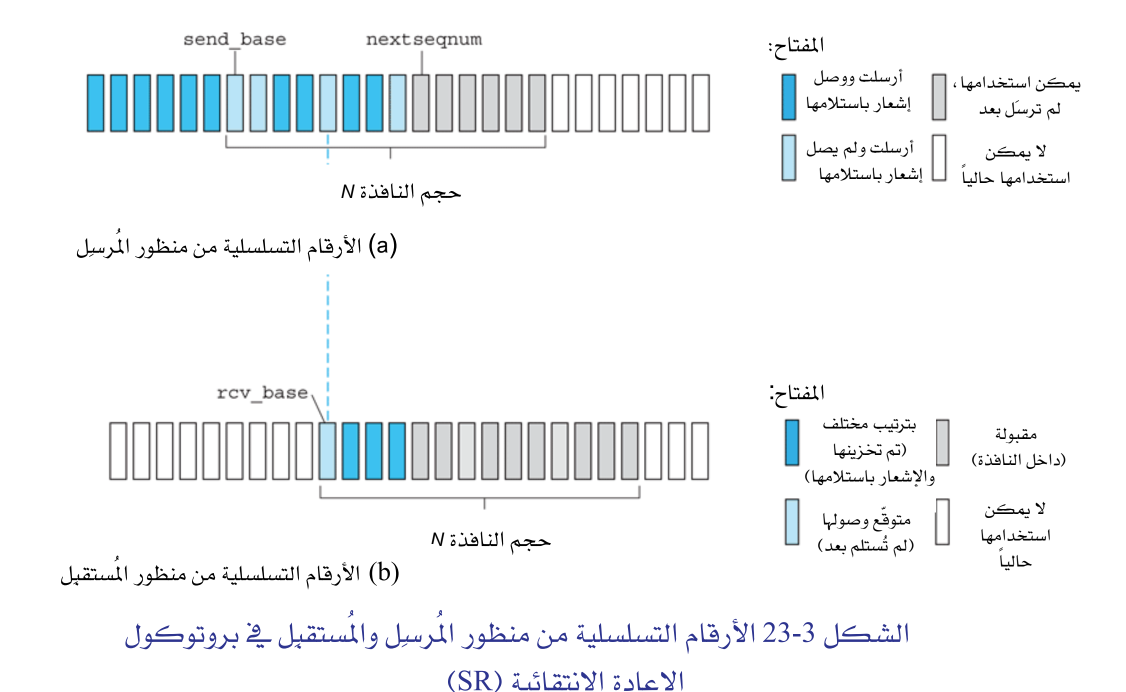
يوفر بروتوكول GBN للمُرسِل إمكانية "ملء خط الأنابيب" في الشكل 3-17 بالرزم، وبهذا يتفادى مشاكل انخفاض معدل استغلال قناة الاتصال التي أشرنا إليها أثناء تناولنا لبروتوكول التوقف والانتظار. ومع ذلك فهناك مواقف يعاني فيها بروتوكول GBN نفسه من مشاكل في الأداء، وبشكلٍ خاص عندما يكون كلٌّ من حجم النافذة وحاصل الضرب (الحيّز الترددي × التأخير) كبيرين، يمكن أن يكون العديد من الرزم في خط الأنابيب على الوصلة في نفس الوقت. أي خطأ في أي حزمة يمكن أن ينتج عنه قيام بروتوكول GBN بإعادة إرسال عدد كبير من الرزم — العديد منها بدون مبرر حيث إنها كانت قد وصلت صحيحة من قبل. مع زيادة احتمال حدوث أخطاء في البيانات أثناء انتقالها عبر القناة، يمكن أن يصبح خط الأنابيب مملوءاً بتلك الرزم المعاد إرسالها بدون داعٍ. في سيناريو إملاء الرسالة على الهاتف الذي تناولناه آنفا في الجزء 3-4-1، تخيل أنه في كل مرة يحدث فيها تشويش على كلمة واحدة نقوم بإعادة إملاء الكلمات الألف التالية لها (بافتراض أن حجم النافذة 1000 كلمة). لاشك أن عملية إملاء الرسالة ستتأخر كثيراً بسبب تكرار الكلمات التي يعاد إرسالها بلا داعٍ.

كما يبدو من الاسم، يتفادى بروتوكول "الإعادة الانتقائية" (SR) أي إعادة إرسال غير ضرورية من المُرسِل، حيث يُعاد فقط إرسال تلك الرزم التي يشتبه المُرسِل في حدوث مشكلة في وصولها لدى المُستقبِل (أي وصلت بخطأ في البيانات أو فُقدت). تتطلب إعادة الإرسال الفردية (عند الحاجة) تلك أن يقوم المُستقبِل بإشعار المُرسِل باستلام الرزم التي تصل صحيحة كلٍّ على حدة بغض النظر عن ترتيبها. مرة أخرى سنستخدم نافذة حجمها N لوضع حد أقصى لعدد الرزم المنتظرة في خط الأنابيب بدون إشعار استلام. غير أنه بخلاف بروتوكول GBN، يكون المُرسِل قد تلقّى إشعارات استلام لبعض تلك الرزم الموجودة في النافذة. يوضح الشكل 3-23

فضاء الأرقام التسلسلية من منظور كلٍّ من المُرسِل والمُستقبِل، كما يبين الشكل 3-24 تفاصيل الإجراءات المختلفة التي تُتخذ من قِبَل المُرسِل في بروتوكول SR.

يقوم المُستقبِل في بروتوكول SR بالإشعار باستلام كل رزمة تصله بشكلٍ صحيح سواء كانت بالترتيب السليم أم لا. يتم تخزين الرزم التي تصل بغير الترتيب السليم بشكلٍ مؤقت لدى المُستقبِل إلى أن تصل أي رزم مفقودة (أي الرزم بأرقام تسلسلية أقل)، وعندئذٍ يتم توصيل جملة الرزم بكاملها بالترتيب السليم إلى الطبقة الأعلى. يحدد الشكل 3-25 الإجراءات المختلفة التي يتخذها المُستقبِل في بروتوكول SR، بينما يصور الشكل 3-26 مثالاً لكيفية عمل بروتوكول SR في حالة وجود رزم مفقودة. لاحظ أنه في الشكل 3-26 يقوم المُستقبِل في البداية بتخزين الرزم {3، 4، 5} بشكلٍ مؤقت، ثم يسلّمها سويةً مع الرزمة 2 إلى الطبقة الأعلى عند استلام الرزمة 2 المتأخرة في النهاية.

من المهم ملاحظة أنه في الخطوة 2 من الشكل 3-25، يعيد المُستقبِل إرسال إشعار باستلام (بدلاً من مجرد إهمال) الرزم التي تم استلامها بأرقام تسلسلية معينة أقل من رقم بداية النافذة الحالية. عليك أن تقنع نفسك بأن إشعار الاستلام هذا مطلوب بالفعل. تبعاً لفضاء الأرقام التسلسلية المبين في الشكل 3-23 للمُرسِل والمُستقبِل، على سبيل المثال إذا لم يكن هناك إشعار استلام للرزمة send_base المنتقلة من المُرسِل إلى المُستقبِل، فإن المُرسِل في النهاية سيعيد إرسال الرزمة send_base، بالرغم من أنه واضح (لنا، ولكن ليس للمُرسِل!) أن المُستقبِل قد استلم تلك الرزمة بالفعل. إذا لم يرسل المُستقبِل إشعاراً باستلام تلك الرزمة، فإن نافذة المُرسِل لن تتقدم للأمام أبداً! يوضح هذا المثال سمة مهمة من سمات البروتوكول SR (والعديد من البروتوكولات الأخرى أيضاً). لن يتوافر للمُرسِل والمُستقبِل دائماً نفس وجهة النظر فيما يتعلق بما تم استلامه بشكلٍ صحيح من عدمه، وهذا يعني أنه في بروتوكول SR لن تتطابق نافذتا كلٍّ من المُرسِل والمُستقبِل على الدوام.



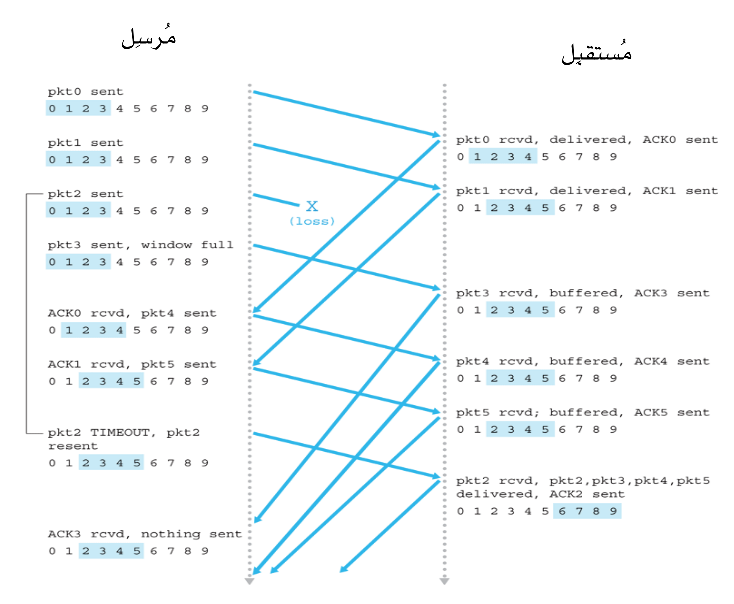
الشكل 3-23 الأرقام التسلسلية من منظور المُرسِل والمُستقبل في بروتوكول الإعادة الانتقائية (SR)

1. استلام البيانات من أعلى: عند استلام البيانات من أعلى، يحدد مُرسِل SR الرقم التسلسلي التالي المتوفر للرزمة. إذا كان الرقم التسلسلي ضمن نافذة المُرسِل، يتم وضع البيانات في رزمة وإرسالها؛ وإلا فإنها إما تخزن بشكل مؤقت لدى المُرسِل أو تعاد إلى الطبقة الأعلى لتحاول إرسالها مرةً أخرى لاحقاً، كما هو الحال في بروتوكول GBN.
2. انقضاء فترة الموقّت: مرةً أخرى تُستخدم الموقّتات هنا أيضاً للحماية ضد فقد الرزم، غير أنه في هذه الحالة يتعين أن يكون لكل رزمة موقّتها المنطقي الخاص بها، حيث يتم إرسال رزمة واحدة فقط عند انقضاء فترة الموقّت. بالإمكان استخدام موقّت مادي واحد لمحاكاة استخدام عدة موقّتات منطقيّة [Varghese 1997].
3. تلقّي إشعار استلام: عندما يتلقى مُرسِل SR إشعار استلام، يؤشر المُرسِل على تلك الرزمة كرزمة تم استلامها بشرط أن تكون في حدود النافذة. إذا كان الرقم التسلسلي للرزمة يساوي رقم قاعدة النافذة send_base يتم تحريك قاعدة النافذة للأمام إلى الرزمة ذات أقل رقم تسلسلي ولم يصل إشعار استلامها بعد. وإذا تحركت النافذة وكانت هناك رزم لم تُرسل ولها أرقام تسلسلية تقع ضمن النافذة الحالية فإنه يتم إرسال تلك الرزم.

الشكل 3-24 الأحداث والإجراءات لدى المُرسِل في بروتوكول الإعادة الانتقائية (SR).

1. استلام رزمة برقم تسلسلي في المدى [rcv_base, rcv_base + N - 1] بشكلٍ صحيح: في هذه الحالة تقع الرزمة المستلَمة ضمن نافذة المُستقبِل، وترسل حزمة إشعار استلام انتقائية إلى المُرسِل، وإذا لم يكن قد تم استلام تلك الرزمة من قبل فإنه يتم تخزينها بشكلٍ مؤقت. وإذا كانت الرزمة لها رقم تسلسلي يساوي رقم قاعدة نافذة المُستقبِل (rcv_base في شكل 3-22)، فإن هذه الرزمة وأي رزم أخرى تم تخزينها مؤقتاً في السابق ولها أرقام تسلسلية متوالية (تبدأ بـ rcv_base) يتم توصيلها إلى الطبقة الأعلى. بعد ذلك يتم تحريك نافذة الاستقبال للأمام بعدد الرزم التي سُلّمت للطبقة الأعلى. كمثال خذ في الاعتبار شكل 3-26. عندما يتم استلام رزمة برقم تسلسلي يساوي rcv_base = 2 ، يمكن توصيل تلك الرزمة مع الرزم 3، و4، و5 للطبقة الأعلى.
2. استلام رزمة برقم تسلسلي في المد [rcv_base - N, rcv-base - 1] بشكلٍ صحيح: في هذه الحالة يجب إرسال إشعار استلام، بالرغم من أن هذه الرزمة تم الإشعار باستلامها من قبل.
3. كل الحالات الأخرى: أهمل الرزمة الواصلة.

الشكل 3-25 الأحداث والإجراءات لدى المُستقبِل في بروتوكول الإعادة الانتقائية (SR).

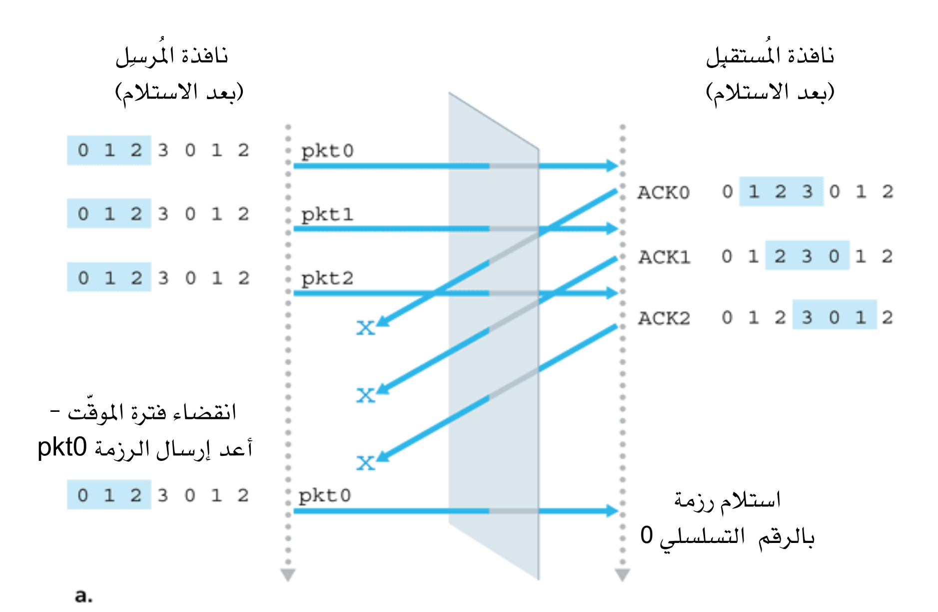


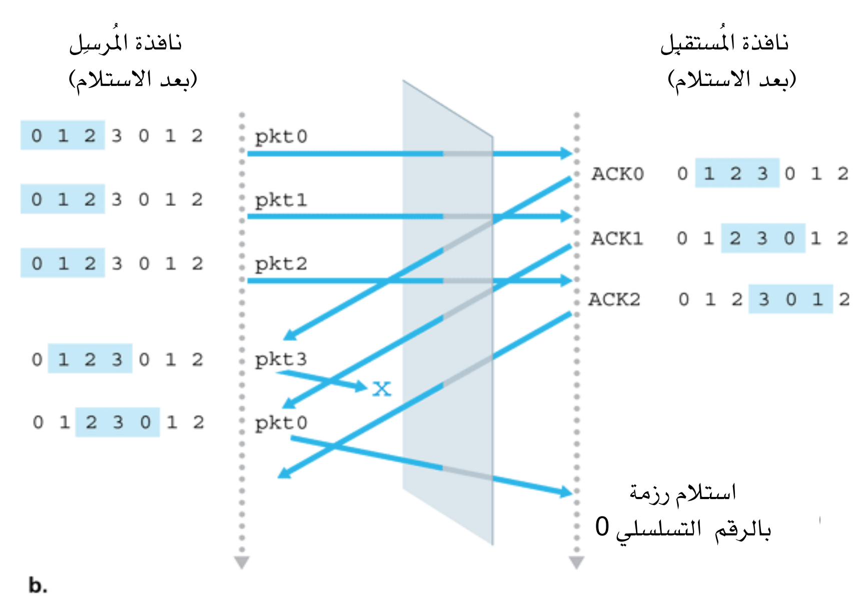
الشكل 3-26 العمل ببروتوكول الإعادة الانتقائية (SR).

يترتب على غياب التزامن بين نافذتي المُرسِل والمُستقبِل نتائج مهمة عندما تواجهنا حقيقة أن مدى الأعداد التسلسلية محدود. خذ بعين الاعتبار ما يمكن أن يحدث على سبيل المثال مع مدى محدود من أربعة أرقام تسلسلية هي {0، 1، 2، 3} وحجم نافذة $N = 3$. افترض أن الرزم من 0 إلى 2 تم إرسالها واستلامها بشكلٍ صحيح وكذلك إرسال إشعارات الاستلام الخاصة بها من قِبَل المُستقبِل. في هذه اللحظة تكون نافذة المُستقبِل تغطي الرزم الرابعة والخامسة والسادسة والتي لها الأرقام التسلسلية 3، 0، 1 على التوالي. لنعتبر السيناريوهين التاليين الآن: في السيناريو الأول والموضح في الشكل 3-27 (a)، تفقد إشعارات الاستلام الخاصة بالرزم الثلاث الأولى ومن ثم يضطر المُرسِل لإعادة إرسال تلك الرزم. وهكذا فإنه في هذه الحالة يتلقى المُستقبِل رزمة برقم تسلسلي 0 — والتي هي نسخة من الرزمة الأولى التي أُرسلت من قبل.

في السيناريو الثاني والموضح في الشكل 3-27 (b)، تصل إشعارات الاستلام للرزم الثلاثة الأولى جميعها بشكلٍ صحيح. وهكذا يدفع المُرسِل بنافذته إلى الأمام ويرسل الرزم الرابعة والخامسة والسادسة بالأرقام التسلسلية {3، 0، 1} على التوالي. تُفقد الرزمة التي تحمل الرقم التسلسلي 3، لكن تصل الرزمة بالرقم التسلسلي 0؛ وهي رزمة تحتوي على بيانات جديدة.

لنأخذ في الاعتبار الآن وجهة نظر المُستقبِل كما في الشكل 3-27، والذي يتضمن ستارة رمزية بين المُرسِل والمُستقبِل، لتذكيرنا بأن المُستقبِل لا يستطيع "رؤية" الإجراءات التي يتخذها المُرسِل. كل ما يلاحظه المُستقبِل هو سلسلة الرسائل التي يتسلمها من القناة ويرسلها إلى القناة. من وجهة نظر المُستقبِل السيناريوهان في الشكل 3-27 متماثلان، فليس لديه طريقة للتمييز ما بين إعادة إرسال الرزمة الأولى والإرسال الأصلي للرزمة الخامسة. واضح أن حجم النافذة الذي يقل بـ 1 عن عدد الأرقام التسلسلية الممكنة لن يعمل بشكلٍ صحيح. ولكن إلى أي حد يحتاج حجم النافذة لأن يكون صغيراً ؟ يُطلب منك في تمرين في نهاية هذا الفصل إثبات أنه في حالة بروتوكول SR يجب أن يكون حجم النافذة أقل من أو يساوي نصف عدد الأرقام التسلسلية الممكنة.





الشكل 3-27 معضلة المُستقبِل في بروتوكول الإعادة الانتقائية (SR): هل الرزمة الواصلة رزمة جديدة أم إعادة إرسال؟

ستجد في موقع الويب المصاحب لهذا الكتاب برنامج جافا صغير يحاكي تشغيل بروتوكول SR. حاول إجراء نفس التجارب التي قمت بها باستخدم بروتوكول GBN لترى ما إذا كانت النتائج تتوافق مع توقعاتك.

بهذا نكون قد أكملنا مناقشتنا لبروتوكولات النقل الموثوق للبيانات. لقد غطّينا كمية كبيرة من المادّة العلمية! واستعرضنا العديد من الآليات التي توفر — مجتمعةً — نقلاً موثوقاً للبيانات. يلخص الجدول 3-1 تلك الآليات. والآن وقد رأينا كل هذه الآليات وهي تعمل وأصبح بوسعنا رؤية "الصورة الكبيرة"، نحثك على مراجعة هذا الجزء ثانيةً لترى كيف تم إضافة تلك الآليات الواحدة تلو الأخرى لتغطي نماذج عملية بمستويات متزايدة من التعقيد لقناة الاتصال التي تربط المُرسِل بالمُستقبِل أو لتحسين أداء البروتوكولات.

دعنا ننهي استعراضنا لبروتوكولات النقل الموثوق للبيانات بتناول الفرضية المتبقية في نموذج قناة اتصالنا التحتية. تذكّر أننا افترضنا أن الرزم لا يمكن إعادة ترتيبها ثانيةً ضمن قناة الاتصال بين المُرسِل والمُستقبِل. تعتبر هذه فرضية معقولة عموماً عندما يكون المُرسِل مربوطاً بالمُستقبِل بواسطة وصلة مادية واحدة. ولكن عندما تكون "القناة" الموصّلة بين الاثنين شبكة، يمكن أن يحدث إعادة ترتيب للرزم. من مظاهر إعادة ترتيب الرزم أن نسخاً قديمة من رزمة برقم تسلسلي أو رقم إشعار استلام x يمكن أن تظهر، رغم أنه لا نافذة المُرسِل ولا نافذة المُستقبِل تتضمن x. مع إعادة ترتيب الرزم يمكن اعتبار قناة الاتصال ببساطة ومن حيث المبدأ كمكان يخزّن الرزم بشكلٍ مؤقت ثم يبعث بها بشكلٍ عفوي في أي لحظة في المُستقبِل. نظراً لأن الأرقام التسلسلية قد تستعمل ثانيةً فإنه ينبغي توخي بعض الحذر في التعامل مع مثل تلك الرزم المزدوجة. تتلخص الطريقة المتبعة في الواقع العملي في تجنب استخدام رقم تسلسلي x مرة ثانية إلا بعد أن يتأكد المُرسِل من أن الرزم المُرسَلة سابقاً بالرقم التسلسلي x لم تعد موجودة في الشبكة. يتم ذلك بافتراض أن الرزمة لا تستطيع "العيش" في الشبكة لأطول من مدة زمنية قصوى ثابتة. تفترض امتدادات بروتوكول TCP للشبكات عمراً أقصى للرزمة قيمته حوالى 3 دقائق [RFC 1323]. يصف [Sunshine 1978] طريقة لاستعمال الأرقام التسلسلية تتفادى تماماً مشاكل إعادة الترتيب.

الآلية	الاستخدام، ملاحظات
المجموع التدقيقي	يُستخدم لاكتشاف أخطاء البتات التي تطرأ على الرزمة المُرسَلة أثناء انتقالها.
الموقّت	يُستخدم لإعادة إرسال رزمة إثر انقضاء فترة تأخير محددة، ربما بسبب فقد الرزمة (أو إشعار استلامها) على القناة. نظراً لأن انقضاء الفترة يمكن أن يحدث عندما تكون الرزمة قد تأخرت فقط دون أن تُفقد (انقضاء قبل الأوان)، أو عندما تكون الرزمة قد استُلمت من قِبَل المُستقبل ولكن إشعار استلامها المبعوث من المُستقبل إلى المُرسِل قد فقد، فإنه يمكن بهذا الأسلوب أن يتلقّى المُستقبل نسخاً مزدوجة من نفس الرزمة.
الأرقام التسلسلية للرزم	تُستخدم للترقيم التسلسلي لرزم البيانات التي تتدفّق من المُرسِل إلى المُستقبل. تسمح الفجوات التي قد تحدث في أرقام الرزم التي يتم استلامها للمُستقبل باكتشاف فقد الرزم. يسمح وصول الرزم بأرقام تسلسلية مكرّرة للمُستقبل باكتشاف النسخ المكرّرة من رزمة.
إشعارات الاستلام	يتم إرسالها من المُستقبل إلى المُرسِل لإخباره بأن رزمة أو مجموعة رزم قد تم استلامها بشكلٍ صحيح. غالباً ما تحمل الإشعارات الرقم التسلسلي للرزمة أو الرزم المراد الإشعار باستلامها. قد تكون تلك الإشعارات فردية أو تراكمية حسب البروتوكول المستخدم.
إشعارات الاستلام السلبية	يتم إرسالها من المُستقبل إلى المُرسِل لإخباره بأن رزمة لم يتم استلامها بشكلٍ صحيح. غالباً ما تحمل الإشعارات السلبية الرقم التسلسلي للرزمة المراد الإشعار بعدم استلامها بشكلٍ صحيح.
النافذة، خط الأنابيب	تستخدم للحَدّ من نشاط المُرسِل بحيث يرسل فقط رزماً بأرقام تسلسلية تقع ضمن مدى محدد (داخل النافذة). بالسماح بإرسال عدة رزم بدون انتظار إشعار باستلامها، يمكن تحسين معدل استغلال المُرسِل مقارنةً ببروتوكول التوقف والانتظار. سنرى بعد قليل أن مقاس النافذة يمكن ضبطه على أساس قدرة المُستقبل على استقبال الرسائل و تخزينها مؤقتاً عنده، أو على مستوى الازدحام في الشبكة، أو على كليهما.

الجدول 3-1 ملخص بآليات النقل الموثوق للبيانات واستخداماتها.

5-3 بروتوكول النقل التوصيلي: TCP

سنرى في هذا الجزء أنه لتوفير نقل موثوق للبيانات، يعتمد بروتوكول TCP على العديد من المبادئ الأساسية التي تناولناها في الجزء السابق، بما في ذلك

اكتشاف الأخطاء، وإعادة الإرسال، وإشعارات الاستلام التراكمية، والموقّتات، وحقول ترويسة الرزمة الخاصة بالأرقام التسلسلية للرزم وأرقام إشعارات الاستلام. تم تعريف بروتوكول TCP من خلال RFC 2581 وRFC 2018 وRFC 1323 و RFC 1122 وRFC 793.

تاريخ حالة (Case History)

فينتون سيرف، وروبرت كاهن، وبروتوكول TCP/IP

في أوائل السبعينيات بدأت شبكات تحويل الرزم في الانتشار، حيث كانت ARPAnet (سلف الإنترنت) مجرد واحدة من الشبكات الموجودة في ذلك الوقت. كان لكل شبكة من تلك الشبكات بروتوكولاتها الخاصة بها. تنبه الباحثان فينتوني سيرف وروبرت كاهن إلى أهمية ربط تلك الشبكات معاً، واخترعا بروتوكولاً للتوصيل بين الشبكات أطلقا عليه اسم TCP/IP (بروتوكول التحكم في الإرسال/بروتوكول الإنترنت). رغم أن سيرف وكاهن كانا في البداية يعتبران هذا البروتوكول كياناً واحداً، إلا أنه تم فصله فيما بعد إلى جزأين يعملان بشكلٍ مستقل هما TCP وIP. في مايو 1974 نشر سيرف وكاهن بحثاً عن بروتوكول TCP/IP في مجلة IEEE لتقنية الاتصالات [Cerf 1974].

لقد ابتُكر بروتوكول TCP/IP – والذي يُعتبر اليوم لُحمة الإنترنت وسداتها – قبل ظهور الحاسب الشخصي ومحطات العمل وانتشار الإيثرنت وتقنيات الشبكة المحلية الأخرى وقبل الويب ونقل الصوت المستمر والدردشة الإلكترونية. لقد أدرك سيرف وكاهن الحاجة إلى بروتوكول لربط الشبكات يوفر من ناحية دعماً واسعاً لتطبيقات لم تحدد بعد، ومن ناحية أخرى يسمح للمضيفات وبروتوكولات ربط البيانات الاختيارية بالعمل معاً على ما يرام.

في عام 2004 نال سيرف وكاهن جائزة Turing من منظمة ACM، والتي تعتبر بمثابة "جائزة نوبل في مجال الحاسبات" تقديراً لعملهما الرائد في تربيط الشبكات (internetworking)، بما في ذلك تصميم وإنجاز بروتوكولات الاتصال الأساسية للإنترنت TCP/IP، ولريادتهما الملهمة في مجال ربط الشبكات.

3-5-1 توصيلة بروتوكول TCP

يقال عن بروتوكول TCP أنه مبني على توصيلة، ذلك لأنه قبل أن تستطيع عملية تطبيق (application process) البدء في إرسال بيانات لعملية أخرى يتعين على العمليتين أولاً إجراء "مصافحة" بينهما. يتم خلال تلك المصافحة تبادل قطع بيانات تمهيدية لتحديد القيم الأولية للمتغيرات الخاصة بنقل البيانات بينهما. كجزء من إجراءات إنشاء التوصيلة في بروتوكول TCP، يقوم كلٌّ من الطرفين بوضع القيم الأوّلية للعديد من متغيرات الحالة (state variables) المتعلقة بتوصيلة البروتوكول (والتي سنتناول الكثير منها في هذا الجزء وفي الجزء 3-7).

جدير بالذكر أن "توصيلة" TCP التي نعنيها هنا ليست دائرة TDM أو FDM من طرف إلى طرف كما في شبكات تحويل الدوائر. كما أنها ليست أيضاً دائرة افتراضية (راجع الفصل الأول)، حيث توجد حالة التوصيلة بالكامل في النظامين الطرفيين. ولأن بروتوكول TCP يجري تشغيله فقط في الأنظمة الطرفية، وليس في الكيانات المتوسطة داخل الشبكة (كالموجّهات ومحوّلات طبقة ربط البيانات)، فإن كيانات الشبكة المتوسطة لا تحتفظ بحالة توصيلة TCP. تكون الموجّهات المتوسّطة في الواقع غافلة تماماً عن توصيلات بروتوكول TCP، فتلك الموجّهات ترى وحدات بيانات وليس توصيلات.

توفر توصيلة TCP خدمة اتصال كامل الازدواج في الاتجاهين (full-duplex)، ففي وجود توصيلة TCP بين العملية A على مضيف ما والعملية B على مضيف آخر، فإن بيانات طبقة التطبيقات يكون بوسعها التدفق من العملية A إلى العملية B في نفس الوقت الذي تتدفق فيه بيانات طبقة التطبيقات من عملية B إلى عملية A. كما أن توصيلة TCP هي دائماً من نقطة إلى نقطة (point-to-point)، بمعنى أنها تكون بين مُرسِل واحد ومُستقبِل واحد. أي أن الاتصال بعدة جهات في نفس الوقت (multicasting) (انظر الجزء 4-7) – حيث تتدفق البيانات من مُرسِل واحد إلى عدة مُستقبلين في عملية إرسال واحدة – غير ممكن في بروتوكول TCP.

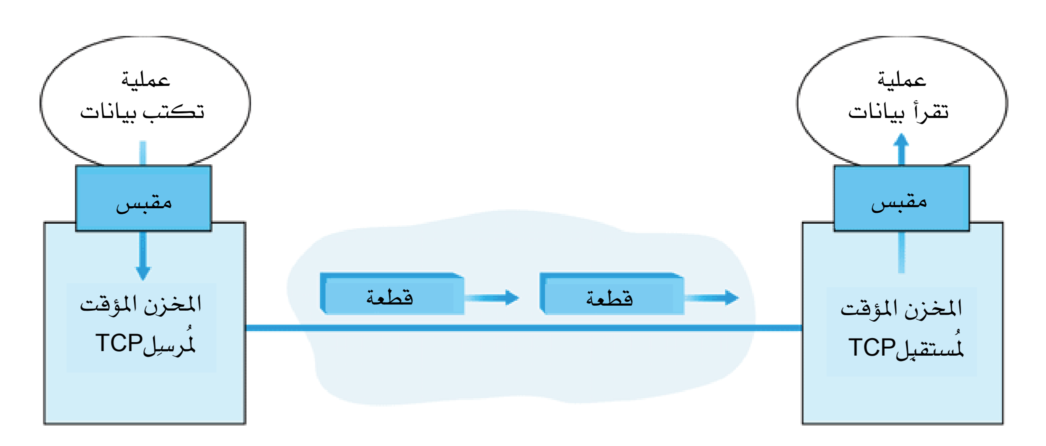
دعنا الآن نلقي نظرة على الكيفية التي يتم بها إنشاء توصيلة TCP. افترض أن عملية يجري تشغيلها على مضيفٍ ما وتود بدء توصيلة مع عملية أخرى على مضيفٍ آخر. تذكّر أن العملية التي تبدأ التوصيلة تُدعى عملية الزبون، بينما يطلق على العملية الأخرى عملية الخادم. في البداية تخبر عملية تطبيق الزبون طبقة النقل على الزبون بأنها تريد إنشاء توصيلة مع عملية الخادم. تذكّر من الجزء 2-7 أن برنامج زبون بلغة جافا يقوم بذلك بإصدار الأمر:

```
Socket clientSocket = new Socket("hostname", portNumber);
```

حيث hostname هو اسم الخادم وportNumber هو رقم المنفذ الذي يميّز العملية على الخادم. تمضي طبقة النقل على الزبون بعد ذلك في إنشاء توصيلة TCP مع بروتوكول TCP على الخادم. في نهاية هذا الجزء سنناقش بشيء من التفصيل إجراءات إنشاء توصيلة TCP. يكفي الآن معرفة أن الزبون يرسل قطعة بيانات TCP خاصة، فيرد الخادم بقطعة بيانات TCP خاصة ثانية، ويرد الزبون أخيراً بقطعة خاصة ثالثة. القطعتان الأوليان لا تحملان أي بيانات لطبقة التطبيقات بينما قد تحمل القطعة الثالثة بيانات لطبقة التطبيقات. نظراً لأنه يتم إرسال ثلاث قطع بيانات بين المضيفين، غالباً ما يطلق على هذا الإجراء لإنشاء توصيلة "إجراء المصافحة الثلاثية".

بمجرد إنشاء توصيلة TCP يصبح بوسع عمليتي التطبيق إرسال البيانات إلى بعضهما البعض. دعنا نأخذ في الاعتبار إرسال البيانات من عملية الزبون إلى عملية الخادم. تقوم عملية الزبون بتمرير سيل البيانات عبر المقبس (بوابة العملية)، كما سبق وصفه في الجزء 2-7. بمجرد مرور البيانات من خلال البوابة تصبح تحت تصرف بروتوكول TCP الذي يجري تشغيله على مضيف الزبون. كما هو موضح في الشكل 3-28، يوجّه TCP تلك البيانات إلى المخزن المؤقت للإرسال، وهو أحد المخازن المؤقتة التي تم إعدادها أثناء عملية المصافحة الثلاثية. من حين لآخر يلتقط نظام TCP كتلة بيانات من المخزن المؤقت لدى المُرسِل ليقوم ببثها. بشكلٍ مثير للانتباه، تعتبر مواصفات بروتوكول TCP [RFC-793] متساهلة جداً فيما يتعلق بتحديد متى يقوم TCP على المُرسِل بإرسال البيانات الموجودة في المخزن المؤقت

لديه، حيث تنص على أنه على البروتوكول أن "يرسل تلك البيانات على شكل قطع في الوقت الذي يراه مناسباً". الحد الأقصى من البيانات الذي يمكن تناوله ووضعه في قطعة يحكمه الحجم الأقصى للقطعة MSS، والذي يتم تحديده أولاً بتعيين الطول الأقصى لإطار طبقة ربط البيانات الذي يمكن لمضيف الإرسال المحلي بثه (وهو ما يسمى بوحدة الإرسال القصوى MTU، بعد ذلك تُحدّد MSS بحيث يمكن استيعاب قطعة بيانات TCP (بعد تغليفها في وحدة بيانات IP) في إطار واحد من إطارات طبقة ربط البيانات. القيم المشهورة للمتغير MTU هي 1,460 بايتاً، 536 بايتاً، و512 بايتاً. تم اقتراح بعض الطرق لتعيين قيمة MTU للمسار — أي إطار طبقة ربط البيانات الأكبر الذي يمكن أن يرسل على كل الوصلات التي ستُستخدم من المصدر إلى الوجهة [RFC 1191]، ومن ثم تحديد قيمة MSS على أساسه. لاحظ أن قيمة MSS هي أقصى كمية لبيانات طبقة التطبيقات في القطعة، وليست الحجم الأقصى لقطعة بيانات TCP والتي تشمل علاوةً على ذلك ترويسة القطعة وحقولها الأخرى. (قد تتسبب المصطلحات هنا في بعض الخلط، ولكن علينا أن نتعايش مع ذلك نظراً للتغلغل الشديد لتلك المصطلحات في أدبيات الشبكات والإنترنت).



الشكل 3-28 مخازن TCP المؤقتة للإرسال والاستقبال.

يضيف بروتوكول TCP لكل كتلة من بيانات الزبون ترويسة TCP الخاصة بها لتكوين قطعة بيانات TCP، ثم يدفع بتك القطع لأسفل إلى طبقة الشبكة، حيث تغلّف كل قطعة على حدة لتكوين وحدات بيانات طبقة الشبكة. يتم إرسال وحدات البيانات تلك خلال الشبكة. عندما يتسلم بروتوكول TCP على مضيف الوجهة قطعة TCP، توضع البيانات المستخلصة من القطعة في مخزن الاستقبال المؤقت التابع لتوصيلة TCP، كما هو موضح في الشكل 3-28، حيث يقرؤها تطبيق الوجهة من ذلك المخزن. لكل جانب من جانبي الاتصال مخزن مؤقت خاص به لكلٍ من الإرسال والاستقبال. يمكنك تشغيل برنامج جافا الخاص بضبط التدفق وذلك بزيارة الموقع http://www.awl.com/kurose-rose والذي يشرح باستخدام الصور المتحركة عمل المخازن المؤقتة للإرسال والاستقبال.

نرى من هذا العرض أن توصيلة TCP تشتمل على مخازن مؤقتة لبيانات المستخدم ومتغيرات ومقبس توصيل لعملية تطبيق على مضيف، بالإضافة إلى مجموعة أخرى من المخازن والمتغيرات ومقبس للعملية المناظرة على المضيف الآخر. كما ذكرنا سابقاً لا يتم تخصيص أي مخازن مؤقتة أو متغيرات تتعلق بتوصيلة TCP في الكيانات المتوسطة بالشبكة (كالموجّهات والمحوّلات والمكرّرات (repeaters) والموجودة بين المضيفين المتصلين.

3-5-2 صيغة قطعة بيانات TCP

بعد أن ألقينا نظرة سريعة على توصيلة TCP دعنا نفحص تركيب قطعة TCP. تتألف قطعة TCP من حقول ترويسة وحقل بيانات (الحمل الآجر) يضم حقل البيانات كتلة بيانات التطبيق المُرسَلة. كما ذكرنا أعلاه يكون الحجم الأقصى لحقل البيانات في القطعة محدوداً بالمتغير MSS. عندما يرسل بروتوكول TCP ملفاً كبيراً، كصورة تمثل جزءاً من صفحة ويب، يقوم البروتوكول عادةً بتجزيء الملف إلى كتل حجم كلٍّ منها MSS (باستثناء الكتلة الأخيرة التي غالباً ما تكون أقل من MSS). غير أن التطبيقات التفاعلية غالباً ما ترسل كتل بيانات أصغر حجماً من MSS. على سبيل المثال في تطبيقات الدخول على الحاسبات عن بعد، مثل Telnet،

غالباً ما يكون حقل البيانات في قطعة TCP بايتاً واحداً فقط. ونظراً لأن ترويسة قطعة TCP تتكون عادةً من 20 بايتاً (أطول من ترويسة قطعة UDP بـ 12 بايتاً) فإن قطع TCP التي يرسلها تطبيق Telnet قد يقتصر طولها على 21 بايتاً فقط.

يبين الشكل 3-29 صيغة قطعة بيانات بروتوكول TCP. كما هو الحال في بروتوكول UDP تتكون ترويسة القطعة من حقول لرقم منفذ المصدر ورقم منفذ الوجهة، والتي تستخدم في تجميع البيانات من تطبيقات الطبقة الأعلى وتوزيعها عليها. كما في UDP تتضمن الترويسة أيضاً حقل المجموع التدقيقي. وعلاوة على ذلك تتضمن ترويسة قطعة بيانات TCP الحقول الإضافية التالية:

32 بتاً

رقم منفذ المصدر | رقم منفذ الوجهة
الرقم التسلسلي
رقم إشعار الاستلام
طول الترويسة | غير مستخدم | URG | ACK | PSH | RST | SYN | FIN | نافذة المستقبل
المجموع التدقيقي للإنترنت | مؤشر البيانات المستعجلة
الخيارات
البيانات

الشكل 3-29 صيغة قطعة بيانات بروتوكول TCP.

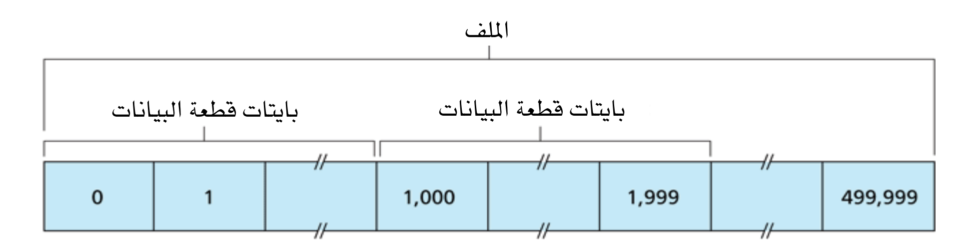
- حقل الرقم التسلسلي للقطعة بطول 32 بتاً وحقل رقم إشعارالاستلام بطول 32 بتاً، والتي يستخدمها مُرسِل ومُستقبِل TCP لتوفير خدمة نقل موثوق للبيانات كما سيتم تفصيله لاحقاً.
- حقل نافذة المُستقبِل بطول 16 بتاً والمستخدم في ضبط التدفق. وسنرى بعد قليل أنه يستخدم لبيان عدد البايتات التي يمكن للمُستقبِل أن يقبلها.
- حقل طول الترويسة بطول 4 بتات ويحدد طول ترويسة قطعة بيانات TCP بالكلمات (words) (الكلمة = 32 بتاً). يمكن أن يكون طول ترويسة TCP متغيراً بسبب وجود حقل خيارات TCP (TCP options). غالباً ما يكون حقل الخيارات خالياً (أي غير مستخدم)، وفي هذه الحالة يكون طول ترويسة TCP هو الطول المعتاد أي 20 بايتاً).
- حقل الخيارات الاختياري، وهو حقل متغير الطول يُستخدم عندما يقوم المُرسِل والمُستقبِل بالتفاوض بخصوص الحجم الأقصى للقطعة MSS أو كمعامل تكبير لمقاس النافذة للاستخدام مع الشبكات عالية السرعة. تم أيضاً تعريف خيار تضمين خاتم الوقت (time stamp). راجع RFC 854 وRFC 1323 للمزيد من التفاصيل.
- حقل الأعلام (أو المؤشرات) ويضم 6 بتات. تستخدم بت إشعار الاستلام (ACK) للإشارة إلى أن القيمة الموجودة في حقل رقم إشعار الاستلام هي قيمة حقيقية؛ أي أن القطعة تتضمن إشعاراً باستلام قطعة قد وصلت بنجاح. تُستخدم البتات RST وSYN وFIN لبدء وإنهاء توصيلة TCP كما سنبين في نهاية هذا الجزء. عند اختيار القيمة 1 للبت PSH، يكون على المُستقبِل تمرير البيانات الواصلة إلى الطبقة الأعلى فوراً. وأخيراً تُستخدم البت URG للإشارة إلى أن هذه القطعة تتضمن بيانات قد علّمها كيان الطبقة الأعلى في جانب المُرسِل كبيانات "مستعجلة" (urgent). يشير حقل مؤشر البيانات المستعجلة (بطول 16 بتاً) إلى موقع البايت الأخيرة في تلك البيانات. يتعين على بروتوكول TCP إخبار كيان الطبقة الأعلى على جانب المُستقبِل عند وجود بيانات مستعجلة مُرسَلة، وإرسال مؤشر يحدد نهاية تلك البيانات

المستعجلة في القطعة. (في الواقع العملي لا تُستخدم بتات الأعلام PSH وURG ولا حقل مؤشر البيانات المستعجلة، ومع ذلك فنحن نذكرها هنا لاستكمال الصورة).

الأرقام التسلسلية لقطع البيانات وأرقام إشعارات الاستلام

يُعدّ الرقم التسلسلي للقطعة ورقم إشعار الاستلام اثنين من أهم حقول ترويسة قطعة بيانات TCP، حيث يمثّلان جزءاً أساسياً من خدمة TCP للنقل الموثوق للبيانات. لكن قبل مناقشة كيفية استخدام هذين الحقلين في تحقيق نقل موثوق للبيانات، دعنا نوضّح أولاً ما الذي يضعه بروتوكول TCP في هذين الحقلين بالضبط.

ينظر بروتوكول TCP للبيانات على أنها سَيل غير مهيكل ولكنه مرتب من البايتات. يعكس استعمال البروتوكول للأرقام التسلسلية وجهة النظر هذه، حيث تمثل تلك الأرقام سلسلة البايتات المُرسَلة وليس سلسلة قطع البيانات المُرسَلة. وبالتالي فإن الرقم التسلسلي الذي تحمله قطعة بيانات هو رقم البايتات التسلسلي لأول بايت بيانات في القطعة. لنأخذ مثالاً. افترض أن عمليةً على المضيف A تريد إرسال سيلٍ من البيانات إلى عملية على المضيف B عبر توصيلة TCP. سيقوم بروتوكول TCP على المضيف A بترقيم كل بايت في سيل البيانات ضمنياً. افترض أن سيل البيانات يتألف من ملف يضم 500000 بايتاً، وأن الحجم الأقصى للقطعة MSS هو 1000 بايت، وأن البايت الأولى في سيل البيانات قد أعطيت الرقم 0. كما هو مبين في الشكل 3-30، يكوّن بروتوكول TCP 500 قطعة من سيل البيانات. وتُعطى القطعة الأولى الرقم التسلسلي 0، والقطعة الثانية الرقم التسلسلي 1000، والقطعة الثالثة الرقم التسلسلي 2000، وهكذا. يوضع كل رقم تسلسلي في حقل الرقم التسلسلي في ترويسة قطعة بيانات TCP المناظرة.



الشكل 3-30 تقسيم بيانات ملف إلى قطع بيانات TCP.

دعنا الآن نأخذ في الاعتبار أرقام إشعارات الاستلام. هذه أصعب بعض الشيء من الأرقام التسلسلية لقطع البيانات. تذكّر أن بروتوكول TCP يعتمد أسلوب الإرسال كامل الازدواج، ولذا فإنه في الوقت الذي يرسل المضيف A البيانات إلى المضيف B يمكنه أيضاً استلام البيانات من المضيف B (كجزء من توصيلة TCP نفسها). تحمل كل قطعة بيانات تصل من المضيف B رقماً تسلسلياً للبيانات التي تتدفق من B إلى A. رقم إشعار الاستلام الذي يضمّنه المضيف A في قطعة بياناته التي يرسلها هو الرقم التسلسلي لبايت البيانات التالية التي يتوقع وصولها من المضيف B. من المفيد النظر إلى بعض الأمثلة لفهم ماذا يجري هنا. لنفترض أن المضيف A قد تسلّم من B كل البايتات المرقمة من 0 إلى 535، وأن A على وشك إرسال قطعة بيانات إلى B. ينتظر المضيف A وصول بايت رقم 536 وكل البايتات التي تليها في سيل البيانات لدى المضيف B. وبالتالي يضع المضيف A الرقم 536 في حقل رقم إشعار الاستلام بقطعة البيانات التي يرسلها إلى B.

وكمثالٍ آخر افترض أن المضيف A قد تسلّم قطعة بيانات واحدة من المضيف B تحتوي على البايتات من 0 إلى 535 وقطعة أخرى تحتوي على البايتات من 900 إلى 1000. لسبب ما لم يتسلّم المضيف A بعد البايتات من 536 إلى 899. في هذا المثال لا يزال المضيف A ينتظر البايت 536 (وما بعدها) لكي يتمكن من إعادة تركيب سلسلة البيانات من A عند وصولها كاملةً لديه. وعليه فإن القطعة التالية من A التي سيرسلها إلى B ستحتوي على 536 في حقل رقم إشعار الاستلام. ونظراً لأن بروتوكول TCP يرسل إشعارات استلام للبايتات لغاية أول بايت مفتقَد ضمن سلسلة البيانات الجاري استقبالها، يقال: إن TCP يستخدم إشعارات استلام تراكمية.

يثير هذا المثال الأخير نقطة مهمة ولكنها دقيقة. افترض أن المضيف A استلم القطعة الثالثة (أي البايتات من 900 إلى 1000) قبل استلام القطعة الثانية (أي البايتات من 536 إلى 899)! ماذا يمكن للمضيف عمله عندما يتسلم قطع بيانات بغير ترتيبها السليم من خلال توصيلة TCP؟ بشكلٍ مثير للانتباه لا تفرض طلبات التعليقات (RFCs) الخاصة ببروتوكول TCP أي قواعد هنا وإنما تترك القرار لمبرمجي النسخة المعنية من البروتوكول. هناك خياران أساسيان: إما أن (1) يهمل المُستقبِل فوراً قطع البيانات الواصلة بغير الترتيب السليم (كما ناقشنا سابقاً، مما يسهم في تبسيط تصميم المُستقبِل) أو (2) يحتفظ المُستقبِل بقطع البيانات الواصلة بغير الترتيب السليم وينتظر وصول القطع المفتقدة لملء الفجوات في البيانات التي يتم استلامها. واضح أن الخيار الأخير أكثر كفاءة من ناحية استغلال الحيّز الترددي للشبكة، وهو الخيار المعمول به في الواقع.

لاحظ أننا في الشكل 3-30 افترضنا أن الرقم التسلسلي الأولي هو 0. في الواقع، يختار كلا الجانبين من توصيلة TCP ذلك الرقم الأول بشكلٍ عشوائي، وذلك لتقليل احتمال وجود قطعة ما في الشبكة من توصيلة سابقة تم إنهاؤها بالفعل بين نفس المضيفين واعتبارها كقطعة صحيحة في توصيلة جديدة بين نفس المضيفين (وتصادف أيضاً أنهما يستعملان نفس أرقام المنافذ المستخدمة في التوصيلة السابقة) [Sunshine 1978].

بروتوكول Telnet : مثال عن الأرقام التسلسلية وأرقام إشعارات الاستلام

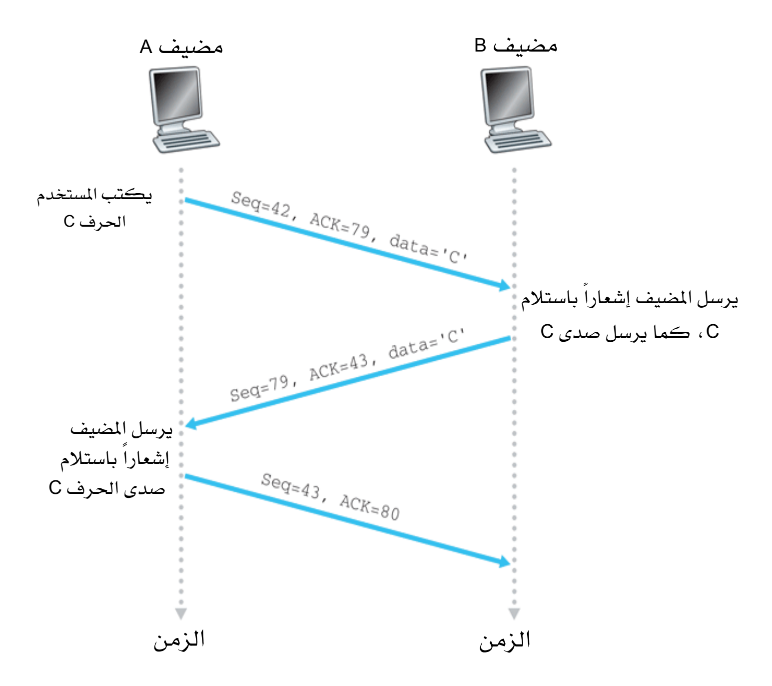
بروتوكول تيلنت (Telnet) هو بروتوكول شهير لطبقة التطبيقات، تم تعريفه بطلب التعليقات RFC 854، ويُستخدم للدخول على الحاسبات عن بعد. يعمل هذا البروتوكول فوق بروتوكول TCP، وهو مصمم للعمل بين أي زوج من المضيفات. خلافاً لمعظم تطبيقات نقل البيانات التي تناولناها في الفصل الثانى، فإن تيلنت يُعتبر تطبيقاً تفاعلياً. نناقش هنا مثال تيلنت لأنه سيوضح بشكلٍ جيد استخدام الأرقام التسلسلية وأرقام إشعارات الاستلام في بروتوكول TCP. جدير بالذكر أن العديد من المستخدمين يفضلون الآن استخدام بروتوكول ssh بدلاً من تيلنت، لأن البيانات

المُرسَلة في توصيلة تيلنت (بما في ذلك كلمات السر!) لا تشفّر، مما يجعل تيلنت عرضةً لهجمات التنصت (كما سنتناوله في الجزء 8-7).

افترض أن المضيف A يبدأ جلسة تيلنت مع المضيف B. ولأن المضيف A هو الذي بدأ الجلسة، فإننا نطلق عليه اسم الزبون، في حين نطلق على المضيف B الخادم. كل حرف يطبعه المستخدم (على الزبون) سيُرسَل إلى المضيف البعيد؛ والذي سيقوم بدوره بإرسال نسخة من كل حرف للعرض على شاشة مستخدم تيلنت. تستخدم طريقة "الصدى" هذه لضمان أن الحروف التي يراها مستخدم تيلنت على شاشته تكون قد تم استقبالها ومعالجتها بالفعل على المضيف البعيد. وعليه فإن كل حرف يكون قد عبر الشبكة مرتين في الفترة من ضغط المستخدِم المفتاح إلى لحظة عرضه على الشاشة أمامه.

افترض الآن أن المستخدم ضغط على حرف ‘C’ على لوحة المفاتيح، وبعدها أخذ يحتسى فنجاناً من القهوة. دعنا نفحص قطع بيانات TCP التي تُرسَل بين الزبون والخادم. كما هو موضح في الشكل 3-31، افترض أن الأرقام التسلسلية الأولى هي 42 و79 للزبون والخادم على التوالي. تذكّر أن الرقم التسلسلي لقطعة هو الرقم التسلسلي للبايت الأول في حقل البيانات بها. وعليه فستحمل القطعة الأولى المُرسَلة من الزبون الرقم التسلسلي 42؛ في حين يكون للقطعة الأولى المُرسَلة من الخادم الرقم التسلسلي 79. تذكّر أن رقم إشعار الاستلام هو الرقم التسلسلي لبايت البيانات التالي الذي ينتظره المضيف. بعد إنشاء وصلة TCP ولكن قبل إرسال أي بيانات، ينتظر الزبون البايت 79 والخادم البايت 42.

كما هو موضح في الشكل 3-31 يتم إرسال ثلاث قطع بيانات. تُرسَل القطعة الأولى من الزبون إلى الخادم وتتضمن في حقل البيانات بها بايت واحد يمثل الحرف المُرسَل ‘C’ حسب تكويد ASCII . تتضمن هذه القطعة أيضاً الرقم 42 في حقل الرقم التسلسلي بها، كما وصفنا آنفاً. لأن الزبون أيضاً لم يتسلم بعد أي بيانات من الخادم، فإن هذه القطعة الأولى ستحمل الرقم 79 في حقل رقم إشعار الاستلام.



الشكل 3- 31 الأرقام التسلسلية وأرقام إشعارات الاستلام أثناء عملية "تيلنت" بسيطة على بروتوكول TCP.

تُرسَل القطعة الثانية من الخادم إلى الزبون لتخدم غرضاً مزدوجاً. أولاً: تُزوّد القطعة الزبون بإشعار باستلام البيانات التي تسلمها الخادم. فبوضع 43 في حقل رقم إشعار الاستلام، يُخبر الخادم الزبون أنه تسلّم كل شيء بنجاح حتى البايت 42 وينتظر الآن البايتات 43 وما بعدها. أما الغرض الثاني لهذه القطعة فهو ترجيع صدى الحرف الواصل 'C' إلى الزبون، ولذا تتضمن القطعة الثانية تمثيل هذا الحرف في حقل البيانات بها. هذه القطعة الثانية لها الرقم التسلسلي 79، أي الرقم التسلسلي الأولي لتدفق البيانات من الخادم إلى الزبون في توصيلة TCP تلك، لأنها تحمل أول بايت بيانات يقوم الخادم بإرساله. لاحظ أن إشعار الاستلام للبيانات من الزبون إلى الخادم تحمله قطعة تحمل بيانات من الخادم إلى الزبون. يطلق على إشعار الاستلام هذا أنه "راكب على الظهر" (piggybacked) على قطعة البيانات من الخادم إلى الزبون.

تُرسَل القطعة الثالثة من الزبون إلى الخادم، وغرضها الوحيد هو إشعار الخادم باستلام البيانات التي أرسلها. (تذكّر أن القطعة الثانية تضمنت بيانات – الحرف 'C' من الخادم إلى الزبون). هذه القطعة لها حقل بيانات فارغ، (أي أن إشعار الاستلام لا "يركب على ظهر" أي بيانات من الزبون إلى الخادم). تحمل القطعة الرقم 80 في حقل رقم إشعار الاستلام لأن الزبون استلم البايتات المُرسَلة إليه حتى بايت 79، ومن ثم ينتظر الآن البايتات 80 وما بعدها. قد ترى من الغريب أن يكون لهذه القطعة رقماً تسلسلياً رغم أنها لا تحتوي على أي بيانات، ولكن نظراً لأن بروتوكول TCP له حقل للرقم التسلسلي، تحتاج القطعة لأن يكون لها رقما تسلسلياً على أي حال.

3-5-3 تقدير وقت رحلة الذهاب والعودة وفترة الموقّت

كما هو الحال في بروتوكول rdt الذي طوّرناه في الجزء 3-4، يستخدم بروتوكول TCP آليات انقضاء فترة الموقّت وإعادة الإرسال للتعافي من فقد قطع البيانات. وبالرغم من بساطة تلك الأساليب من حيث المبدأ، يظهر عدد من الأمور الدقيقة عندما نطبّق آلية انقضاء فترة الموقّت لإعادة الإرسال في بروتوكول حقيقي مثل TCP. لعل أكثر الأسئلة إلحاحًا يتعلق بطول فترات الموقّت المناسبة. واضح أن تلك الفترة يجب أن تكون أكبر من وقت رحلة الذهاب والإياب *RTT* على التوصيلة – أي الوقت من لحظة إرسال قطعة إلى حين وصول إشعار باستلامها– وإلا فقد يحدث إعادة إرسال بدون داعٍ. لكن إلى أي حد أكبر؟ بل كيف نقدر قيمة الوقت *RTT* بدايةً؟ هل نستخدم موقّتاً لكل قطعة لم يصل إشعار باستلامها؟ أسئلةٌ كثيرة! تستند مناقشتنا في هذا الجزء لبروتوكول TCP حسب [Jacobson 1988] وتوصيات فريق عمل هندسة الإنترنت (IETF) الحالية بخصوص إدارة موقّتات TCP [RFC2988].

تقدير وقت رحلة الذهاب والإياب

لنبدأ دراستنا لإدارة الموقّتات في بروتوكول TCP بالنظر في الطريقة التي يتبعها البروتوكول لتقدير وقت رحلة الذهاب والإياب (*RTT*) بين المُرسِل والمُستقبِل. يتم ذلك كالتّالي: يُعرّف وقت عينة *RTT* لقطعة بيانات، والذي نرمز له بـ SampleRTT ، بأنه الوقت المنقضي منذ إرسال القطعة (أي دفعها إلى بروتوكول IP بطبقة الشبكة) إلى وصول إشعار باستلامها. بدلاً من قياس SampleRTT لكل قطعة يتم إرسالها، تكتفي معظم تطبيقات TCP المستخدمة بقياس SampleRTT كعينة مرة واحدة كلما دعت الحاجة. بمعنى أنه عند كل نقطة زمنية، يجري تقدير SampleRTT لقطعة واحدة فقط من القطع التي أُرسِلت ولكن لم يتم الإشعار باستلامها بعد، مما يعطي قيمة جديدة لـ SampleRTT تقريباً مرة كل *RTT*. يلاحظ أيضاً أن TCP لا يقيس أبداً قيمةً SampleRTT لقطعة يعاد إرسالها؛ وإنما يقيسها فقط للقطع التي أرسلت مرة واحدة. (هناك تمرين في نهاية هذا الفصل يطلب منك تبرير ذلك).

واضحٌ أن قيمة SampleRTT ستتفاوت من قطعة إلى أخرى بسبب الازدحام في الموجّهات وتغيّر الأحمال على الأنظمة الطرفية، ولذلك يمكن أن تكون بعض قيم SampleRTT شاذة. لتقدير قيمة نمطية للوقت *RTT*، من الطبيعي حساب نوع من المتوسط لقيم SampleRTT المقاسة. يحتفظ TCP بقيمة متوسطة لقيم SampleRTT تُعرف بـ EstimatedRTT. عند الحصول على قيمة SampleRTT جديدة، يقوم TCP بتحديث تقديره للكمية EstimatedRTT حسب المعادلة التالية:

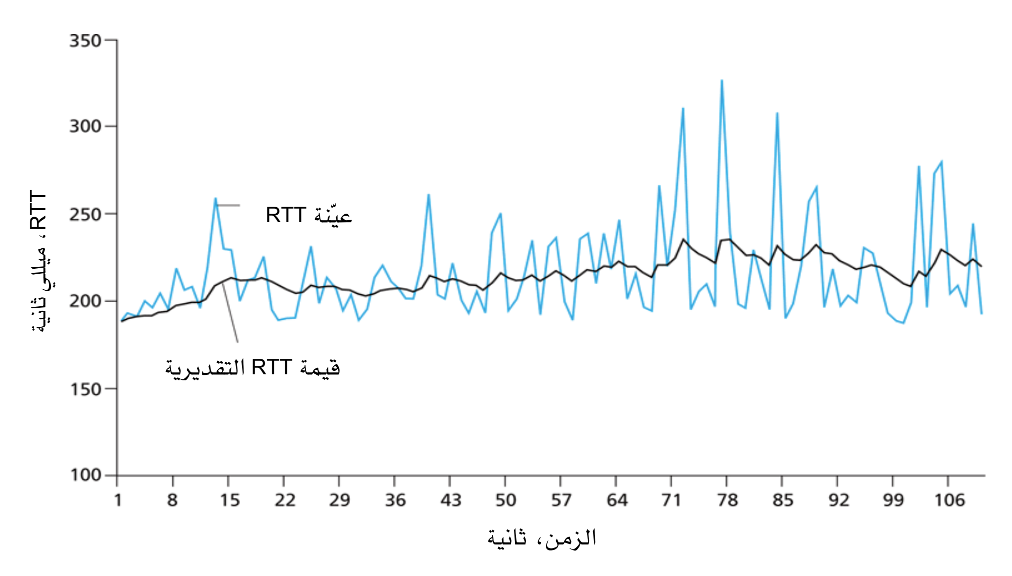
$$\text{EstimatedRTT} = (1 - \alpha) \times \text{EstimatedRTT} + \alpha \times \text{SampleRTT}$$

لاحظ أن تلك المعادلة مكتوبة على شكل تعليمة من تعليمات لغة البرمجة، حيث القيمة الجديدة لـ EstimatedRTT هي تركيبة موزونة من القيمة السابقة لـ EstimatedRTT والقيمة الجديدة لـ SampleRTT. القيمة الموصّى بها لـ α هي 0.125 [RFC 2988]، وفي هذه الحالة تصبح المعادلة:

$$\text{EstimatedRTT} = 0.875 \times \text{EstimatedRTT} + 0.125 \times \text{SampleRTT}$$

لاحظ أن EstimatedRTT هو معدل موزون لقيم SampleRTT. كما نوقش في أحد التمارين في نهاية هذا الفصل، هذا المتوسط الموزون يضع وزناً أكبر للعينات الأخيرة مقارنة بالعينات القديمة. وهذا الأمر طبيعي لأن العينات الأحدث تعكس بشكلٍ أدق وضع الازدحام الحالي في الشبكة. وفي علم الإحصاء يُطلق على مثل هذا المتوسط بالمتوسط المتحرك بأوزان أسّية (exponential weighted moving average (EWMA)). تظهر كلمة "أسّي" في الإسم لأن الوزن المعطى لعينة SampleRTT يضمحل بسرعة أُسّية مع توالي التجديدات. في التمارين سيطلب منك اشتقاق الحد الأسّي في EstimatedRTT.

يبين الشكل 3-32 قيم SampleRTT وقيم EstimatedRTT في حالة α = 0.125 لتوصيلة TCP من المضيف gaia.cs.umass.edu (في مدينة Amherst بولاية Massachusetts)، إلى المضيف fantasia.eurecom.fr (في جنوب فرنسا). واضح أنه تم التخفيف من حدة الاختلافات الكبيرة في قيم SampleRTT من خلال حساب EstimatedRTT.



الشكل 3-32 عينات وقيم *RTT* التقديرية.

بعد الحصول على تقدير لقيمة متوسطة للوقت *RTT*، من المفيد أيضاً التوصل لمقياس لمدى التغير في *RTT*. يعرّف [RFC 2988] التفاوت في *RTT* (أي DevRTT) كتقدير لقيمة التغير الذي يُتوقع أن تنحرف به قيمة العينة SampleRTT عادةً عن قيمة المتوسط EstimatedRTT:

$$DevRTT = (1 - \beta) \times DevRTT + \beta \times | SampleRTT - EstimatedRTT |$$

لاحظ أنّ DevRTT هو متوسط متحرك بأوزان أسّية (EWMA) للفرق بين SampleRTT وEstimatedRTT. إذا كانت قيم SampleRTT تعاني من تفاوتات قليلة، فإن DevRTT سيكون صغيراً؛ وعلى العكس إذا كانت الاختلافات في قيم العينات كبيرة، فإن DevRTT سيكون كبيراً. قيمة β الموصى بها هي 0.25.

ضبط وإدارة فترة موقّت إعادة الإرسال (Timeout Interval)

إذا كانت لدينا قيمة لكلٍّ من EstimatedRTT وDevRTT، فما هي القيمة التي يجب أن نستخدمها لفترة الموقّت في TCP والتي بانقضائها يتم إعادة الإرسال؟ واضح أن تلك الفترة يجب أن تكون أكبر من أو تساوي EstimatedRTT، وإلا فإن عمليات إعادة إرسال ستتم بدون داعٍ. لكن فترة الموقّت لا ينبغي أيضاً أن تكون أكبر بكثير من EstimatedRTT، وإلا فعند فقد قطعة بيانات سيتأخر نظام TCP في إعادة إرسالها ومن ثم يتسبب في تأخيرات كبيرة في نقل البيانات. وعليه فمن المستحب اختيار فترة الموقّت بحيث تزيد قليلاً عن قيمة EstimatedRTT، ويجب أن تكون تلك الزيادة كبيرة عندما يكون هناك الكثير من التقلبات في قيم SampleRTT، وتكون قليلة عندما تكون تلك التقلبات صغيرة. معنى ذلك أن قيمة DevRTT يجب أن تلعب دوراً هنا. تُؤخذ كل هذه الاعتبارات في الحسبان في الطريقة التي يتبعها بروتوكول TCP لتحديد فترة موقّت إعادة الإرسال (TimeoutInterval):

$$TimeoutInterval = EstimatedRTT + 4 \times DevRTT$$

المبادئ في الواقع العملي (Principles in Practice)

يوفر بروتوكول TCP نقلاً موثوقاً للبيانات باستخدام إشعارات الاستلام الإيجابية والموقّتات تقريباً بنفس الطريقة التي درسناها في الجزء 3-4. يرسل TCP إشعاراً باستلام قطع البيانات التي يتسلمها بشكلٍ صحيح، ويعيد إرسال القطع متى غلب على ظنه أن تلك القطع أو إشعارات استلامها قد فُقدت أو وصلت فاسدة. تتضمن بعض إصدارات TCP أيضاً آلية ضمنية لإشعارات الاستلام السلبية (NAK). ففي آلية TCP السريعة لإعادة الإرسال، يؤخذ استلام ثلاثة إشعارات استلام إيجابية مكرّرة لقطعة ما على أنه إشعار استلام سلبي (NAK) ضمني بخصوص القطعة التالية، ومن ثم يؤدي إلى إعادة إرسال تلك القطعة قبل انقضاء فترة الموقّت. يستخدم بروتوكول TCP الأرقام التسلسلية للسماح للمُستقبِل باكتشاف قطع البيانات المفقودة أو المكرّرة. تماماً كما في حالة البروتوكول rdt3.0 للنقل الموثوق للبيانات لن يتمكن بروتوكول TCP بنفسه من أن يحدد على وجه اليقين ما إذا كانت قطعة بيانات (أو إشعار استلامها) قد فُقدت أو فسدت أو تأخرت أكثر مما ينبغي. سيكون لبروتوكول TCP في المُرسِل نفس رد الفعل إزاء كل تلك الاحتمالات: إعادة إرسال القطعة موضع التساؤل.

يستخدم بروتوكول TCP أيضاً أسلوب خط الأنابيب (pipelining) للسماح للمُرسِل بأن يكون لديه عدة قطع بيانات منتظرة في وقت ما، تكون قد أُرسلت ولم تصل إشعارات باستلامها بعد. رأينا في وقت سابق أن هذا الأسلوب يمكن أن يحسّن كثيرا من الطاقة الإنتاجية للجلسة عندما يكون طول قطعة البيانات صغيراً مقارنةً بتأخير رحلة الذهاب والإياب. يتم تحديد العدد المعين من القطع المنتظرة بدون إشعار استلام لدى المُرسِل عن طريق آليات التحكم في الازدحام وضبط التدفق. سنتناول ضبط التدفق في بروتوكول TCP في نهاية هذا الجزء، أما التحكم في الازدحام في TCP فسنناقشه في الجزء 3-7. الآن علينا فقط وببساطة إدراك أن TCP يستخدم أسلوب خط الأنابيب لتحقيق ذلك.

3-5-4 النقل الموثوق للبيانات

تذكّر أن خدمة طبقة شبكة الإنترنت (خدمة IP) غير موثوقة، فهى لا تضمن توصيل وحدات بيانات IP، ولا تضمن توصيل تلك الوحدات بالترتيب السليم، ولا

تضمن سلامة البيانات المستلمة في تلك الوحدات. فمع خدمة IP، يمكن لوحدات البيانات أن تفيض في المخازن المؤقتة بالموجّهات فتُفقد ولا تصل إلى وجهتها أبداً، كما يمكن لتلك الوحدات أن تصل فاسدة بأخطاء في بتاتها (حيث يُقلب الـ 0 إلى 1 والعكس بالعكس). ولأن قطع طبقة النقل يتم نقلها عبر الشبكة بواسطة وحدات بيانات IP تلك، فإن قطع طبقة النقل يمكن أن تعاني من نفس المشاكل أيضاً.

ينشىء بروتوكول TCP خدمة نقل موثوق للبيانات فوق خدمة IP غير الموثوقة التي تعتمد طريقة الجهد الأفضل. تضمن خدمة TCP للنقل الموثوق للبيانات أن سيل البيانات الذي تقرؤه عمليةٌ ما من مخزن TCP المؤقت غير فاسدة، وبدون فجوات، وبدون تكرار، وبالترتيب السليم — أي أنها تتسلم بالضبط نفس سيل البايتات الذي أرسله النظام الطرفي على الجانب الآخر من توصيلة TCP . إذن فكيف يوفر TCP نقلاً موثوقاً للبيانات؟ يتضمن ذلك العديد من المبادئ التي درسناها في الجزء 3-4.

في تطويرنا السابق لأساليب النقل الموثوق للبيانات، كان من السهل من حيث المبدأ افتراض تخصيص موقّت على حِدة لكل قطعة بيانات تم إرسالها ولم يصل بعد إشعار باستلامها. رغم أن هذا رائع نظرياً إلا أن إدارة الموقّتات بهذه الطريقة ستشكل عبئاً كبيراً، ولذا فإن الإجراءات الموصى بها لإدارة موقّتات TCP [RFC 2988] تنص على استعمال موقّت واحد فقط لإعادة الإرسال، حتى لو كان هناك عدة قطع بيانات تم إرسالها ولم يصل إلى المُرسِل الإشعار باستلامها بعد. يتبع بروتوكول TCP الذي نصِفُه في هذا الجزء هذه التوصية باستعمال موقّت واحد.

سنناقش هنا كيف يوفر بروتوكول TCP نقلاً موثوقاً للبيانات في خطوتين تدريجيتين. نقدم أولاً وصفاً مبسطاً بدرجةٍ كبيرة لمُرسِل TCP يستخدم فقط أسلوب المؤقّت للتعافى من فقد قطع البيانات، ثمّ نقدم بعد ذلك وصفاً أكثر كمالاً لمُرسِل يستخدم إشعارات الاستلام المكرّرة بالإضافة إلى انقضاء فترة الموقّت. سنفترض في

المناقشة التالية أن البيانات تُرسَل فقط في اتجاه واحد، من المضيف A إلى المضيف B، وأن المضيف A يرسل ملفاً كبيراً.

يوضح الشكل 3-33 وصفاً مبسطاً للغاية لمُرسِل TCP، حيث نرى أن هناك ثلاثة أحداث رئيسة تتعلق بإرسال وإعادة إرسال البيانات في مُرسِل TCP: وصول البيانات من التطبيق في الطبقة الأعلى، وانقضاء فترة الموقّت، ووصول إشعار استلام. عند وقوع الحدث الرئيس الأول، يتسلم TCP البيانات من التطبيق ويغلّفها في قطعة، ثم يدفع بالقطعة إلى بروتوكول IP في طبقة الشبكة أسفله. لاحظ أن كل قطعة بيانات تتضمن رقماً تسلسلياً هو رقم بايت البيانات الأول في القطعة ضمن سيل البايتات الجاري إرساله، كما سبق وصفه في الجزء 3-5-2. لاحظ أنه إذا لم يكن الموقّت شغالاً لقطعة أخرى، فإن TCP يبدأ الموقّت عندما يدفع بالقطعة إلى IP (من المفيد اعتبار الموقّت مرتبطاً بأقدم قطعة لم يتم وصول إشعار باستلامها بعد). افترض أن فترة الانتهاء لهذا الموقّت هي TimeoutInterval، والتي يتم حسابها بمعلومية EstimatedRTT وDevRTT، كما تقدم وصفه في الجزء 3-5-3.

يتمثل الحدث الرئيس الثاني في انقضاء فترة الموقّت، والذي يستجيب له TCP بإعادة إرسال القطعة التي تسببت في انقضاء فترة الموقّت. بعد ذلك يقوم TCP ببدء تشغيل الموقّت من جديد لقطعة أخرى.

الحدث الرئيس الثالث الذي يجب على مُرسِل TCP التعامل معه هو وصول قطعة تحمل إشعار استلام ACK من المُستقبِل (بتحديد أكثر، قطعة تحتوي على قيمة حقيقية في حقل إشعار الاستلام). عند حدوث ذلك يقارن TCP قيمة إشعار الاستلام y بالمتغير SendBase لديه. لاحظ أن المتغير SendBase الذي يسهم في وصف حالة TCP هو الرقم التسلسلي لأقدم بايت لم يصل إشعار استلام له بعد. (وهكذا يكون SendBase - 1 هو الرقم التسلسلي للبايت الأخير الذي تم التأكد من أنه قد وصل بشكلٍ صحيح وبالترتيب السليم). كما سبق أن أشرنا، يستخدم TCP إشعارات استلام تراكمية، بحيث يُخبر إشعار الاستلام y باستلام كل البايتات التي أُرسلت قبل البايت رقم y. إذا كانت $y >$ SendBase، فإن إشعار الاستلام

الواصل يُخبر باستلام قطعة أو أكثر من قطع لم يتم الإشعار باستلامها سابقاً. وعليه يقوم المُرسِل بتحديث المتغير SendBase لديه، كما أنه يبدأ تشغيل الموقّت من جديد إذا كانت هناك أي قطع متبقية لم يصل إشعار باستلامها.

```
/* Assume sender is not constrained by TCP flow or congestion control, that
data from above is less than MSS in size, and that data transfer is in one
direction only * /

NextSeqNum=InitialSeqNumber
SendBase=InitialSeqNumber

loop (forever) {
    switch (event)
    event: data received from application above
            create TCP segment with sequence number NextSeqNum
            if (timer currently not running)
                 start timer
            pass segment to IP
            NextSeqNum=NextSeqNum+length (data)
            break;

    event: timer timeout
            retransmit not-yet-acknowledged segment with smallest
                    sequence number
            start timer
            break;

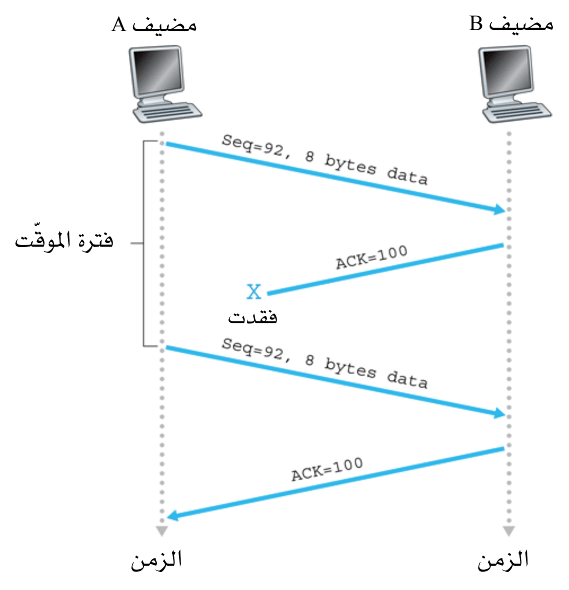
    event: ACK received, with ACK field value of y
            if (y > SendBase) {
                  SendBase=y
                  if (there are currently any not-yet-acknowledged segments)
                        start timer
            }
            break,;

} /* end of loop forever * /
```

الشكل 3-33 مُرسِل مبسّط لبروتوكول TCP.

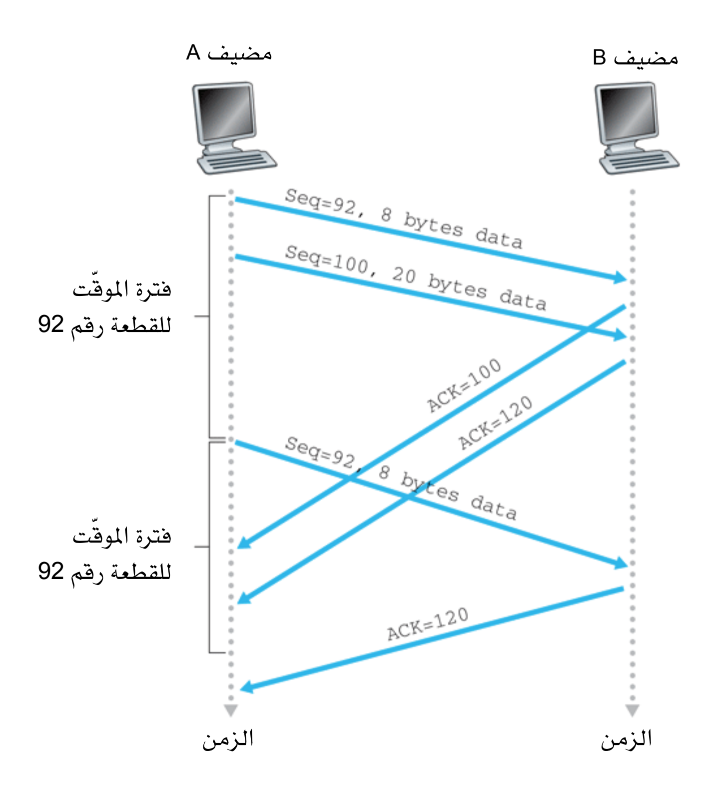
بضعة سيناريوهات هامة

انتهينا للتوّ من وصف مبسّط للغاية لأسلوب TCP في توفير نقل موثوق للبيانات. ولكن حتى هذا النمط المبسّط جداً من TCP غنيٌ بالكثير من الأمور الدقيقة والهامة. ولفهم جيد لكيفية عمل هذا البروتوكول دعنا الآن نتلمّس طريقنا خلال بضعة سيناريوهات مبسّطة. يبين الشكل 3-34 السيناريو الأول، حيث يرسل المضيف A قطعة واحدة للمضيف B. افترض أن هذه القطعة لها الرقم التسلسلي 92 وتتضمن 8 بايتات من البيانات. بعد إرسال تلك القطعة، ينتظر المضيف A قطعة من B تحمل إشعار استلام برقم 100. لنفرض أنه رغم استلام B للقطعة المُرسَلَة من A، إلا أن إشعار الاستلام من B إلى A فُقد في الطريق. في هذه الحالة تنتهي فترة الموقّت، ويقوم المضيف A بإعادة إرسال نفس القطعة. بالطبع عندما يستقبل المضيف B القطعة المعاد إرسالها، سيلاحظ من الرقم التسلسلي للقطعة أنها تحمل بيانات سبق استلامها، ولذا يقوم TCP على المضيف B بإهمال بايتات البيانات في تلك القطعة.



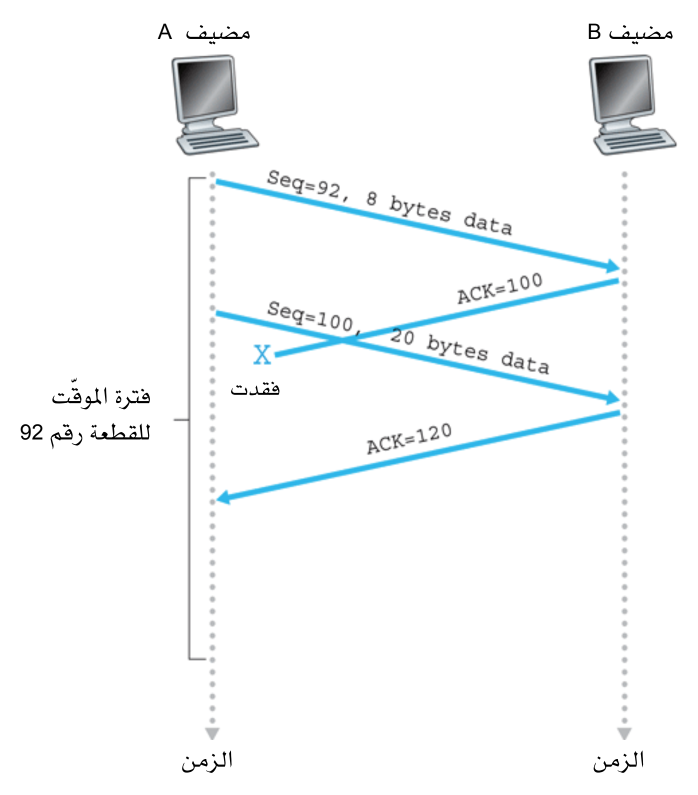
الشكل 3-34 إعادة الإرسال نتيجة فقد إشعار استلام.

في السيناريو الثاني والموضح في الشكل 3-35 يرسل المضيف A قطعتي بيانات الواحدة تلو الأخرى مباشرةً، الأولى لها الرقم التسلسلي 92 وتحمل 8 بايتات من البيانات، والثانية لها الرقم التسلسلي 100 وتحمل 20 بايت من البيانات. افترض أن كل قطعة وصلت سليمة إلى B، وأن B أرسل إشعار استلام منفصل لكل منهما، الإشعار الأول رقمه 100، والثاني رقمه 120. افترض الآن أن أياً من إشعاري الاستلام لم يصل إلى المضيف A قبل انقضاء فترة الموقّت. عند انقضاء فترة الموقّت يقوم المضيف A بإعادة إرسال القطعة الأولى برقم تسلسلي 92 ويبدأ تشغيل الموقّت من جديد. في حالة وصول إشعار استلام القطعة الثانية قبل انقضاء فترة الموقّت الجديدة، فإن القطعة الثانية لن يعاد إرسالها.



الشكل 3-35 لن يعاد إرسال القطعة 100.

في السيناريو الثالث والأخير والموضح في الشكل 3-36 افترض أن المضيف A أرسل القطعتين بالضبط كما في المثال الثاني، وأن إشعار الاستلام الخاص بالقطعة الأولى فقط فُقد في الشبكة، ولكن مباشرةً قبل انقضاء فترة الموقّت تسلّم المضيف A إشعار استلام رقمه 120. عندئذٍ يدرك المضيف A أن المضيف B قد تسلّم كل شيء لغاية البايت 119، ولذا فإن المضيف A لا يعيد إرسال أيٍّ من القطعتين.



الشكل 3-36 يؤدي إشعار الاستلام التراكمي إلى تجنب إعادة إرسال القطعة الأولى.

مضاعفة فترة الموقّت

سنناقش الآن بضعة تعديلات تتضمنها أكثر تطبيقات بروتوكول TCP المستخدمة حالياً. يتعلّق التعديل الأول بطول فترة الموقّت، فحينما تنقضي فترة الموقّت، يعيد TCP إرسال قطعة البيانات التي لم يصل إشعار باستلامها والتي لها

أصغر رقم تسلسلي كما ذكرنا سابقاً. ولكن في كل مرة يقوم البروتوكول بعملية إعادة إرسال، يضبط فترة الموقّت التالية بحيث تكون ضعف قيمتها السابقة، بدلاً من اشتقاقها باستخدام آخر قيم متاحة لـ EstimatedRTT وDevRTT (كما تقدم وصفه في الجزء 3-5-3). على سبيل المثال افترض أن فترة الموقّت TimeoutInterval المرتبطة بأقدم قطعة لم يتم الإشعار باستلامها هي 0.75 ثانية عندما انقضت فترة الموقّت للمرة الأولى. عندئذ سيعيد TCP إرسال تلك القطعة ثم يضبط فترة الموقّت الجديدة عند 1.5 ثانية. إذا انقضت فترة الموقّت مرة أخرى بعد 1.5 ثانية، فإن TCP سيعيد إرسال تلك القطعة من جديد، ثم يضبط فترة الموقّت هذه المرة عند 3.0 ثانية. وهكذا تنمو فترة الموقّت تصاعدياً بعد كل إعادة إرسال. ومع ذلك فحينما يتم بدأ تشغيل الموقّت بعد أيٍّ من الحدثين الآخرين (أي تسلّم بيانات من تطبيق في الطبقة الأعلى أو وصول إشعار استلام) فإن الفترة TimeoutInterval يتم اشتقاقها كالمعتاد من آخر قيم متوفرة لـ EstimatedRTT وDevRTT.

يمثل هذا التعديل شكلاً محدوداً من طرق التحكم في الازدحام. (سنتناول أشكالاً أشمل لوسائل التحكم في الازدحام في الجزء 3-7). غالباً ما تنقضي فترة الموقّت بسبب ازدحام الشبكة، أي بسبب وصول رزم كثيرة إلى واحد (أو أكثر) من صفوف الانتظار في موجّه أو أكثر على المسار بين المصدر والوجهة النهائية، مما يتسبّب في فقد بعض الرزم أو زيادة في تأخيرات الانتظار في الصف. في أوقات الازدحام إذا استمرت مصادر البيانات في إعادة إرسال الرزم المفقودة أو المتأخرة بإصرار، فإن حالة الازدحام قد تزداد سوءاً. بدلاً من ذلك يتصرّف TCP بطريقة أفضل حيث يقوم كل مُرسِل بإعادة الإرسال بعد فترات أطول فأطول. وسنرى عند دراستنا لأسلوب CSMA/CD في الفصل الخامس أن بروتوكول الإيثرنت يستخدم طريقة مماثلة.

الإعادة السريعة للإرسال

من مشاكل أسلوب إعادة الإرسال أن فترة الموقّت يمكن أن تكون طويلة نسبياً، فعند فقد قطعة بيانات، تؤدى فترة الموقّت الطويلة تلك إلى تأخير المُرسِل إرسال الرزمة المفقودة مرة ثانية، وبذلك يزداد التأخير الذي تعانيه الرزمة من طرف إلى طرف. لكن لحسن الحظ غالباً ما يكون بوسع المُرسِل اكتشاف فقد الرزمة قبل انقضاء فترة الموقّت بمدة طويلة وذلك بملاحظة إشعارات الاستلام المكرّرة. إشعار الاستلام المكرّر هو إشعار باستلام قطعة سبق أن تلقى المُرسِل ما يفيد استلامها. لإدراك طبيعة رد المُرسِل على إشعار استلام مكرّر، علينا التفكير في السبب الذي يجعل المُستقبِل يرسل إشعار استلام مكرّر بدايةً. يلخص الجدول 3-2 سياسة مُستقبِل TCP في توليد إشعارات الاستلام [RFC 1122, RFC 2581]. عندما يتلقى مُستقبِل TCP قطعةً لها رقم تسلسلي أكبر من الرقم التسلسلي التالي المتوقع بالترتيب السليم، فإنه يكتشف فجوة في سيل البيانات التي تصله من المُرسِل - أي يكتشف افتقاد (غياب) قطعة. يمكن أن تنتج تلك الفجوة من فقد أو تبديل ترتيب القطع داخل الشبكة. نظراً لأن بروتوكول TCP لا يستخدم أسلوب إشعارات الاستلام السلبية، فإن المُستقبِل لا يستطيع إرسال إشعار استلام سلبي صريح إلى المُرسِل، لكن عوضاً عن ذلك يقوم المُرسِل ببساطة بإعادة الإشعار باستلام آخر بايت بيانات استلمها صحيحة بالترتيب السليم (أي يرسل إشعار استلام مكرّر لها). (لاحظ أن الجدول 3-2 يتضمن الحالة التي لا يهمل فيها المُستقبِل القطع التي تصل بغير الترتيب السليم).

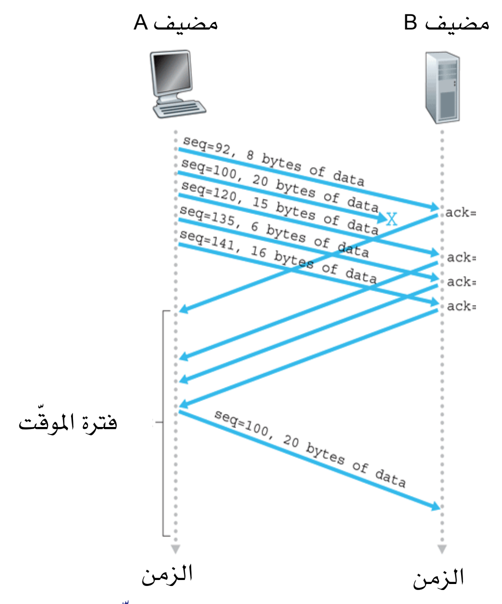
الحدث	الإجراء لدى مُستقبِل TCP
وصلت قطعة بالترتيب السليم وبالرقم التسلسلي المتوقّع. كل البيانات لغاية القطعة بالرقم التسلسلي المتوقّع تم الإشعار باستلامها.	إشعار استلام متأخر. انتظر وصول قطعة أخرى بالترتيب السليم لمدة أقصاها 500 ميللي ثانية. إذا لم تصل القطعة التالية حسب الترتيب السليم خلال تلك الفترة، أرسل إشعار استلام.
وصلت قطعة بالترتيب السليم وبالرقم التسلسلي المتوقّع. توجد قطعة أخرى وصلت بالترتيب السليم تنتظر إرسال إشعار باستلامها.	أرسل إشعار استلام تراكمي واحد في الحال للإشعار باستلام كلا القطعتين الواصلتين بالترتيب السليم.
وصلت قطعة بترتيب غير سليم تحمل رقماً تسلسلياً أكبر من المتوقّع. اكتشاف فجوة.	أرسل حالاً إشعار استلام مكرّر يحمل الرقم التسلسلي للبايت التالية المتوقعة (والتي تمثل الطرف الأدنى للفجوة).
وصلت قطعة تملأ (جزئياً أو كلياً) الفجوة في البيانات التي يجري استقبالها.	أرسل حالاً إشعار استلام، إذا كانت القطعة تبدأ عند الطرف الأدنى للفجوة.

الجدول 3-2 توصيات إصدار إشعارات الاستلام في مُستقبِل TCP [RFC1122; RFC2581].

نظراً لأن المُرسِل يرسل في أغلب الأحيان عدداً كبيراً من القطع، الواحدة تلو الأخرى، فإنه إذا فقدت قطعة واحدة، فمن المحتمل أن يؤدي ذلك إلى إرسال العديد من إشعارات الاستلام المكرّرة الواحد تلو الآخر. إذا استلم مُرسِل TCP ثلاثة إشعارات استلام مكرّرة لنفس قطعة البيانات، سيأخذ ذلك كإشارة إلى أن القطعة التالية للقطعة التي تم الإشعار باستلامها ثلاث مرات قد فقدت. (في التمارين سنناقش لماذا ينتظر المُرسِل تلقّي ثلاثة إشعارات استلام مكرّرة وليس فقط مجرد تلقي إشعار استلام مكرّر واحد). عند تلقّي ثلاثة إشعارات استلام مكرّرة يقوم مُرسِل TCP بإعادة سريعة للإرسال [RFC 2581]، حيث يعيد إرسال القطعة المفقودة قبل انقضاء فترة موقّت تلك القطعة. يُبيّن ذلك في الشكل 3-37 حيث تُفقد القطعة الثانية ثم يُعاد إرسالها قبل أن تنتهي فترة موقّتها. في هذه الحالة يمكن استبدال حدث وصول إشعار الاستلام في الشكل 3-33 بالأوامر التالية للحصول على بروتوكول TCP بإعادة سريعة للإرسال:

```
event: ACK received, with ACK field value of y
        if (y > SendBase) {
                SendBase=y
                if (there are currently any not-yet-acknowledged segments)
                        start timer
        }
        else {
                /* a duplicate ACK for already ACKed segment */
                increment number of duplicate ACKs received for y
                if (number of duplicate ACKS received for y==3){
                        /* TCP fast retransmit */
                        resend segment with sequence number y
        }
        break;
```

لاحظنا في وقت سابق أن العديد من الأمور الدقيقة تنشأ عند تطبيق آلية انقضاء فترة موقّت إعادة الإرسال في بروتوكول فعلي مثل TCP. إن الإجراءات السابقة والتي نتجت كحصيلة لأكثر من 15 عاماً من الخبرة والتجربة مع موقّتات TCP، حريٌ بها أن تقنعك بهذه الحقيقة!



الشكل 3-37 إعادة إرسال قطعة قبل انقضاء فترة الموقّت الخاص بها.

"الرجوع N للوراء" (GBN) أم "الإعادة الانتقائية" (SR)؟

دعنا نختم دراستنا لآليات بروتوكول TCP للتعافي من الأخطاء بطرح السؤال التالي: هل يستخدم بروتوكول TCP أسلوب "الرجوع N للوراء" (GBN) أم أسلوب "الإعادة الانتقائية" (SR)؟ تذكّر أن إشعارات الاستلام في TCP تراكمية، وأن القطع التي تُستلم بشكلٍ صحيح ولكن بغير الترتيب السليم لا يتم الإشعار باستلامها بشكلٍ فردي من قِبَل المُستقبِل. ولذا فكما هو موضح في الشكل 3-33 (انظر أيضاً الشكل 3-19)، يحتاج مُرسِل TCP فقط للاحتفاظ بأصغر رقم تسلسلي للبايت التي أُرسِلت ولم يتم بعد الإشعار باستلامها (SendBase) وكذلك الرقم التسلسلي للبايت التالية التي عليها الدور في الإرسال (NextSeqNum). بهذا المفهوم يشبه TCP كثيراً البروتوكولات بأسلوب "الرجوع N للوراء" (GBN)، غير أن هناك عدة اختلافات واضحة بين بروتوكول TCP وبروتوكولات GBN. العديد من تطبيقات TCP تقوم بالتخزين المؤقت لقطع البيانات التي تصل بشكلٍ صحيح ولكن بترتيب غير سليم [Stevens 1994]. تأمّل أيضاً ما يحدث عندما يقوم المُرسِل بإرسال القطع 1، 2، . . . ، N وكل القطع تصل صحيحة وبالترتيب السليم إلى المُستقبِل. افترض أيضاً أن إشعار الاستلام للرزمة n فُقد، حيث $n < N$، لكن كل إشعارات الاستلام الأخرى لباقي القطع وعددها $(N - 1)$ تصل إلى المُرسِل قبل انقضاء فترة الموقّت. في هذا المثال يعيد بروتوكول GBN إرسال القطعة n وكلّ القطع التالية لها $n + 1$، $n + 2$، . . .، N، وليس فقط القطعة n وحدها. أما بروتوكول TCP فيعيد إرسال قطعة واحدة على الأكثر وهي القطعة n. بل أكثر من ذلك، قد لا يعيد بروتوكول TCP إرسال حتى القطعة n في حالة وصول إشعار استلام القطعة $n + 1$ قبل انقضاء فترة الموقّت للقطعة n.

من التعديلات المقترحة لبروتوكول TCP تعديلٌ يُعرف بالإشعار الانتقائي بالاستلام [RFC 2018]. يسمح هذا التعديل لمُستقبِل TCP بالإشعار باستلام القطع التي تصل بغير الترتيب السليم بشكلٍ انتقائي بدلاً من الاكتفاء فقط بالإشعار باستلام آخر قطعة صحيحة بالترتيب السليم بشكلٍ تراكمي. عند دمج هذه

الطريقة مع أسلوب الإعادة الانتقائية للإرسال – مع تجنب إعادة إرسال القطع التي أُقرّت بشكلٍ انتقائي من قبل المُستقبِل – فإن بروتوكول TCP يبدو كثير الشبه ببروتوكول SR العام الذي سبق أن تناولناه. وهكذا فإن أفضل تصنيف لآلية التعافي من الأخطاء في بروتوكول TCP قد يكون اعتباره كهجين ما بين بروتوكولي GBN وSR.

3-5-5 ضبط التدفق (Flow Control)

تذكّر أن المضيفين على جانبي توصيلة TCP يخصصان حيّز تخزين مؤقت لاستقبال قطع البيانات الواصلة. عندما تتسلم توصيلة TCP بايتات صحيحة وبالترتيب السليم تضعها في مخزن الاستقبال المؤقت. تقوم عملية التطبيق المناظرة بقراءة البيانات من ذلك المخزن، ولكن ليس بالضرورة في نفس لحظة وصول البيانات. في الواقع قد يكون التطبيق المقصود مشغولاً بمهمّة أخرى وقد لا يحاول قراءة البيانات حتى بعد فترة طويلة من وصولها. إذا كان التطبيق بطيئاً نسبياً في قراءة البيانات، يمكن بسهولة أن يتسبب المُرسِل في فيضان المخزن المؤقت لدى المُستقبِل إذا قام بإرسال بيانات أكثر من اللازم بسرعة أكبر من اللازم.

يوفر بروتوكول TCP للتطبيقات التي تستخدمه خدمة لضبط التدفق لتجنب احتمال تسبب المُرسِل في فيضان المخزن المؤقت لدى المُستقبِل. وعليه فضبط التدفق هو خدمة لمواءمة السرعة – أي مواءمة السرعة التي يرسل بها المُرسِل البيانات للسرعة التي يقوم بها التطبيق المُستقبِل بقراءة تلك البيانات على الطرف الآخر من التوصيلة. كما ذكرنا سابقاً يمكن أيضاً أن يُحِد بروتوكول TCP من قدرة المُرسِل على الإرسال بسبب الازدحام في شبكة IP. يعرف هذا الشكل من التحكم في المُرسِل بالتحكم في الازدحام وهو موضوع آخر سنتناوله بالتفصيل في الجزأين 3-6 و3-7. ورغم أن الإجراءات التي تُتخذ للسيطرة على الازدحام تشبه تلك المتبعة في ضبط التدفق (أي الحد من سرعة إرسال المُرسِل للبيانات)، فإن تلك الإجراءات تُتخذ لأسباب مختلفة جداً في الحالتين. قد يستخدم الكثير من الباحثين المصطلحين بشكلٍ متبادل، غير أنه يجدر بالقارئ الواعي أن يميز بينهما. دعنا

نناقش الآن كيف يوفر TCP خدمة ضبط التدفق. لكي نرى الغابة من خلال الأشجار، سنفترض في كافة أنحاء هذا الجزء أن المُستقبِل في بروتوكول TCP المستخدم يقوم بإهمال قطع البيانات التي تصل بترتيب غير سليم.

يوفر بروتوكول TCP ضبط التدفق بجعل المُرسِل يحتفظ بمتغير يطلق عليه نافذة المُستقبِل. من حيث المبدأ يُستخدم هذا المتغير لإعطاء المُرسِل فكرة عن حيّز التخزين المؤقت المتاح لدى المُستقبِل. نظراً لأن الإرسال في بروتوكول TCP كامل الازدواج في الاتجاهين (Full-duplex)، يحتفظ المُرسِل في كلٍ من جانبي التوصيلة بنافذة مُستقبِل مميزة. دعنا نتحرّى استخدام متغير نافذة المُستقبِل ضمن سياق إرسال الملفات. افترض أن المضيف A يرسل ملفاً كبيراً إلى المضيف B على توصيلة TCP. يخصص المضيف B حيّز تخزين مؤقت لتلك التوصيلة، سنرمز لحجمه بالرمز RcvBuffer. من حين لآخر تقوم عملية التطبيق في المضيف B بالقراءة من ذلك المخزن المؤقت. دعنا نعرّف المتغيرات التالية:

- LastByteRead: رقم آخِر بايت في سيل البيانات قامت عملية التطبيق على المضيف B بقراءته.
- LastByteRcvd: رقم آخِر بايت في سيل البيانات وصل من الشبكة وتم وضعه في المخزن المؤقت على المضيف B.

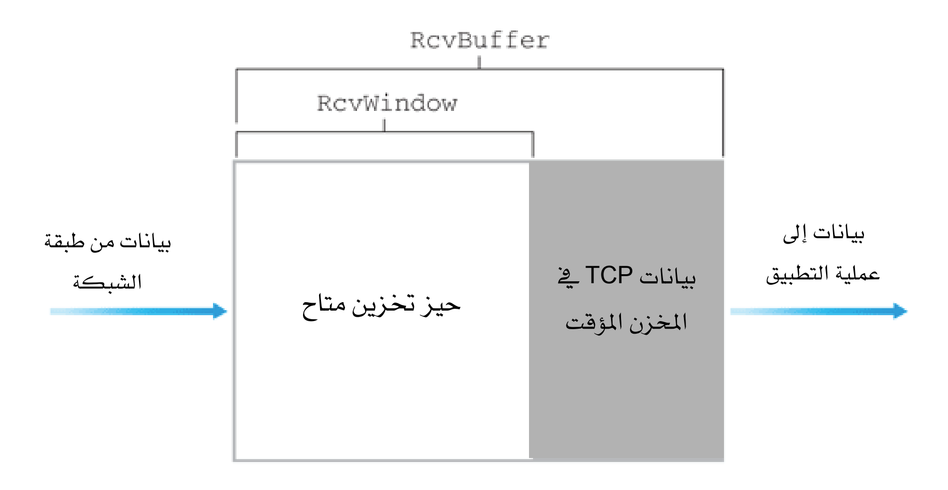
نظراً لأن TCP لا يسمح بفيضان المخزن المؤقت لدى المُستقبِل، ينبغي أن يكون:

$$\text{LastByteRcvd} - \text{LastByteRead} \leq \text{RcvBuffer}$$

يتم ضبط قيمة نافذة المُستقبِل بحيث تساوي سعة التخزين المتاحة في مخزن الاستقبال المؤقت:

$$\text{RcvWindow} = \text{RcvBuffer} - [\text{LastByteRcvd} - \text{LastByteRead}]$$

نظراً لأن حيّز التخزين المتاح يتغير مع الوقت، تتغير قيمة RcvWindow ديناميكياً كما هو موضح في الشكل 3-38.



الشكل 3-38 نافذة المُستقبِل (RcvWindow) والمخزن المؤقت لدى المستقبل (RcvBuffer)

كيف تُستخدم توصيلة TCP المتغير RcvWindow لتوفير خدمة ضبط التدفق؟ يخبر المضيف B المضيف A عن حيّز التخزين المتاح في مخزن الاستقبال لديه بوضع القيمة الحالية لـ RcvWindow في حقل نافذة المُستقبِل بكل قطعة بيانات يرسلها إلى المضيف A. لاحظ أنه لكي يتمكن B من ذلك، عليه أن يتابع التغيرات الحاصلة في عدّة متغيرات تتعلق بتوصيلة TCP. في البداية يضع المضيف B قيمة المتغير RcvWindow بحيث تساوي RcvBuffer.

يقوم المضيف A بدوره بمتابعة متغيرين هما LastByteSent وLastByteAcked واللذين يدل اسمهما بوضوح على وظيفتيهما. لاحظ أن حاصل طرح هذين المتغيرين، أي (LastByteSent - LastByteAcked)، يمثل كمية البيانات التي أرسلها A ولم يتلق من B إشعاراً باستلامها بعد. بالإبقاء على كمية البيانات التي لم يتم الإشعار باستلامها أقل من قيمة RcvWindow، يمكن للمضيف A أن يطمئن أنّه لن يتسبب في فيضان المخزن المؤقت في المضيف B. وعليه، يسعى المضيف A لضمان تحقق الشرط التالي طوال مدة التوصيلة:

$$\text{LastByteSent} - \text{LastByteAcked} \leq \text{RcvWindow}$$

يعاني هذا النظام من مشكلة فنية بسيطة. لاكتشاف هذه المشكلة افترض أن المخزن المؤقت على المضيف B ممتلئ، أي أن RcvWindow = 0. بعد إخطار المضيف B للمضيف A أن RcvWindow = 0، افترض أيضاً أن B ليس لديه ما يرسله إلى A. تصور ما يمكن أن يحدث الآن. بينما عملية التطبيق في B تفرّغ المخزن المؤقت، لا يرسل TCP قطع بيانات جديدة بقيم RcvWindow جديدة من B إلى A. في الواقع إن TCP على المضيف B يرسل قطعاً إلى A فقط إذا كان لديه بيانات يود إرسالها أو إشعار استلام يود إرساله. وبالتالي لن تتاح الفرصة أبداً للمضيف B لإخبار المضيف A أن انفراجاً قد حدث وأن هناك فراغاً متاحاً الآن للتخزين على المضيف B — لذا يظل المضيف A معاقاً ولا يمكنه إرسال المزيد من البيانات! لحل هذه المشكلة، تتطلّب مواصفات TCP من المضيف A مواصلة إرسال قطع بيانات تضم بايتاً واحدة عندما يكون حجم نافذة المُستقبِل على المضيف B صفراً. سيقوم B بدوره بإرسال إشعارات استلام لتلك القطع. بمرور الوقت سيوجد مكان خالٍ في المخزن المؤقت على B وستحتوي إشعارات الاستلام على قيمة غير صفرية للمتغير RcvWindow. يتضمن الموقع المصاحب لهذا الكتاب على الإنترنت http://www.awl.com/kurose-ross برنامج جافا تفاعلي يصوّر عمل نافذة المُستقبِل في بروتوكول TCP. بعد أن استعرضنا خدمة ضبط التدفق في بروتوكول TCP، نذكر سريعاً هنا أن بروتوكول UDP لا يوفر تلك الخدمة. لفهم تلك القضية خذ في الاعتبار إرسال سلسلة من قطع بيانات UDP من عملية على المضيف A إلى عملية على المضيف B. في تطبيق نمطي لبروتوكول UDP سيضع البروتوكول القطع في مخزن مؤقت محدود الحجم يسبق المقبس المناظر للعملية التي تستقبل البيانات (والذي يمثل البوابة إلى العملية). تقرأ العملية قطعة واحدة كاملة في كل مرة من ذلك المخزن المؤقت. فإذا كانت العملية لا تقرأ القطع من المخزن بالسرعة الكافية، سيفيض المخزن بالبيانات وتُفقد القطع.

6-5-3 إدارة توصيلة TCP

في هذا الجزء من الكتاب سنلقي نظرة أكثر تفحصاً على كيفية إنشاء وإنهاء توصيلات TCP. هذا الموضوع قد لا يبدو مثيراً بدرجة كبيرة، إلا أنه مهم - فإنشاء توصيلة TCP يمكن أن يضيف بشكلٍ ملحوظ إلى التأخيرات المحسوسة (على سبيل المثال، عند تصفح الويب). علاوة على ذلك فإن العديد من الهجمات الأكثر شيوعاً على الشبكات – بما في ذلك هجوم فيضان SYN ذائع الصيت – تستغل نقاط الضعف في إدارة توصيلات TCP. دعنا أولاً نلقي نظرة على كيفية إنشاء توصيلة TCP. افترض أن عملية يجري تشغيلها على مضيف ما (زبون) تريد بدء توصيلة مع عملية أخرى على مضيف آخر (خادم). تقوم عملية تطبيق الزبون أولاً بإخبار بروتوكول TCP على الزبون بأنها تريد إنشاء توصيلة مع عملية على الخادم. بعد ذلك يمضي بروتوكول TCP على الزبون في إنشاء توصيلة TCP مع بروتوكول TCP على الخادم بالطريقة التالية:

- خطوة 1: يرسل بروتوكول TCP في ناحية الزبون أولاً قطعة TCP خاصة إلى بروتوكول TCP على الخادم. لا تحتوي تلك القطعة على أية بيانات من طبقة التطبيقات، ولكن أحد بتات الأعلام في ترويسة القطعة (انظر الشكل 3-29)، وهي البت SYN، تكون لها القيمة 1. لهذا السبب تعرف هذه القطعة الخاصّة باسم قطعة SYN. بالإضافة لذلك يختار الزبون رقماً تسلسلياً عشوائياً (client_isn) ويضع هذا الرقم في حقل الرقم التسلسلي لقطعة SYN الأولي. يتم تغليف هذه القطعة لتكوين وحدة بيانات IP وترسل إلى الخادم. استحوذت عملية الاختيار العشوائي الجيد لرقم client_isn للزبون على اهتمام كبير، وذلك لتفادي بعض الهجمات الأمنية على الشبكة [CERT 2001-09].
- خطوة 2: بمجرد وصول وحدة بيانات IP التي تتضمن قطعة SYN إلى مضيف الخادم (على افتراض أنها ستصل!)، ينتزع الخادم قطعة SYN من وحدة البيانات، ويخصّص مخازن TCP المؤقتة ومتغيرات توصيلة TCP، ثم يرسل قطعة "منح توصيلة" إلى الخادم TCP (سنرى في الفصل الثامن أن

تخصيص تلك المخازن المؤقتة والمتغيرات للتوصيلة قبل إكمال الخطوة الثالثة من خطوات المصافحة الثلاثية تجعل بروتوكول TCP عرضةً لهجوم لحجب الخدمة يُعرف بفيضان SYN). لا تتضمن قطعة "منح توصيلة" هي الأخرى أي بيانات من طبقة التطبيقات، غير أنها تحتوي على ثلاث معلومات مهمة في ترويسة القطعة:

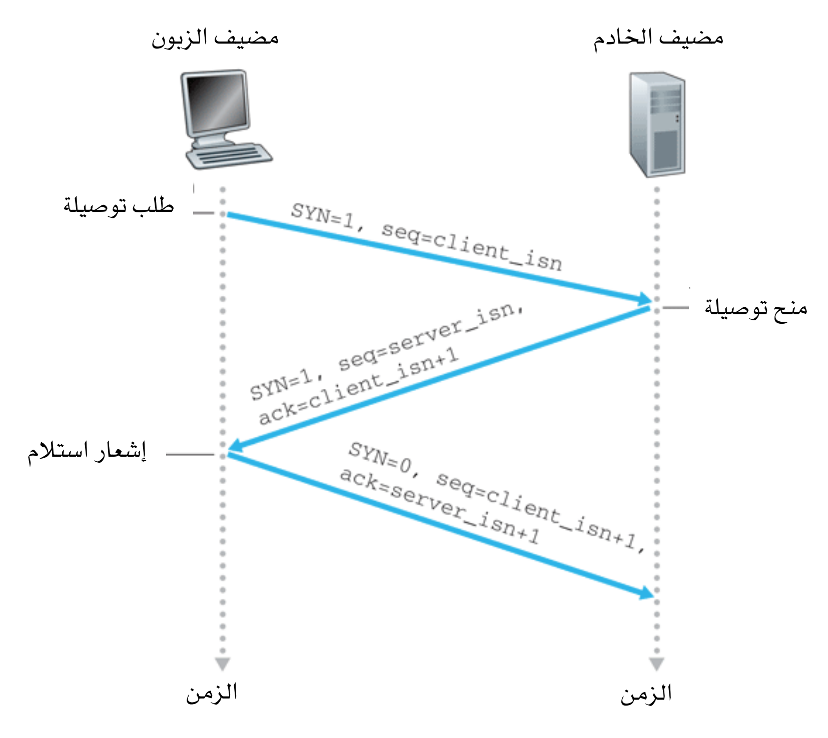
أولاً: البت SYN وقيمتها 1، ثانياً: حقل رقم إشعار الاستلام في ترويسة قطعة TCP وتكون له القيمة client_isn+1، وأخيراً: يختار الخادم رقمه التسلسلي الأول الخاص به server_isn ويضع قيمته في حقل الرقم التسلسلي بترويسة قطعة TCP.

في الواقع، لسان حال قطعة "منح توصيلة" هذه يقول: "استلمت رزمتك من نوع SYN لبدء توصيلة برقمك التسلسلي client_isn، وأوافق على إنشاء تلك التوصيلة، ورقمي التسلسلي الأولي هو server_isn". تعرف قطعة "منح توصيلة" باسم قطعة SYNACK.

- خطوة 3: عند استلام قطعة SYNACK يخصص الزبون أيضاً المخازن المؤقتة والمتغيرات الخاصة بالتوصيلة، ويرسل مضيف الزبون بعد ذلك إلى مضيف الخادم قطعة أخرى لإشعار الخادم بوصول قطعة "منح توصيلة" (يقوم الزبون بذلك بوضع القيمة server_isn+1 في حقل رقم إشعار الاستلام في ترويسة قطعة TCP). يكون للبت SYN القيمة صفر، حيث إن التوصيلة تم إنشاؤها. هذه المرحلة الثالثة من مراحل المصافحة الثلاثية يمكن أن تحمل بيانات من الزبون إلى الخادم في حقل payload للقطعة.

بمجرد الانتهاء من تلك الخطوات الثلاث، يمكن أن يُرسل مضيفا الخادم والزبون قطعاً تحتوي على بيانات إلى بعضهما البعض. في كل تلك القطع المُستقبلية تكون قيمة البت SYN صفراً. لاحظ أنه لإنشاء توصيلة، تبادل المضيفان ثلاثة رزم كما يبين الشكل 3-39، ولهذا السبب غالباً ما يطلق على هذا الإجراء لإنشاء التوصيلة اسم "المصافحة الثلاثية". يتم استكشاف عدة سمات لمصافحة TCP الثلاثية في التمارين في نهاية الفصل (مثلاً لماذا يحتاج الأمر إلى أرقام تسلسلية أولية؟

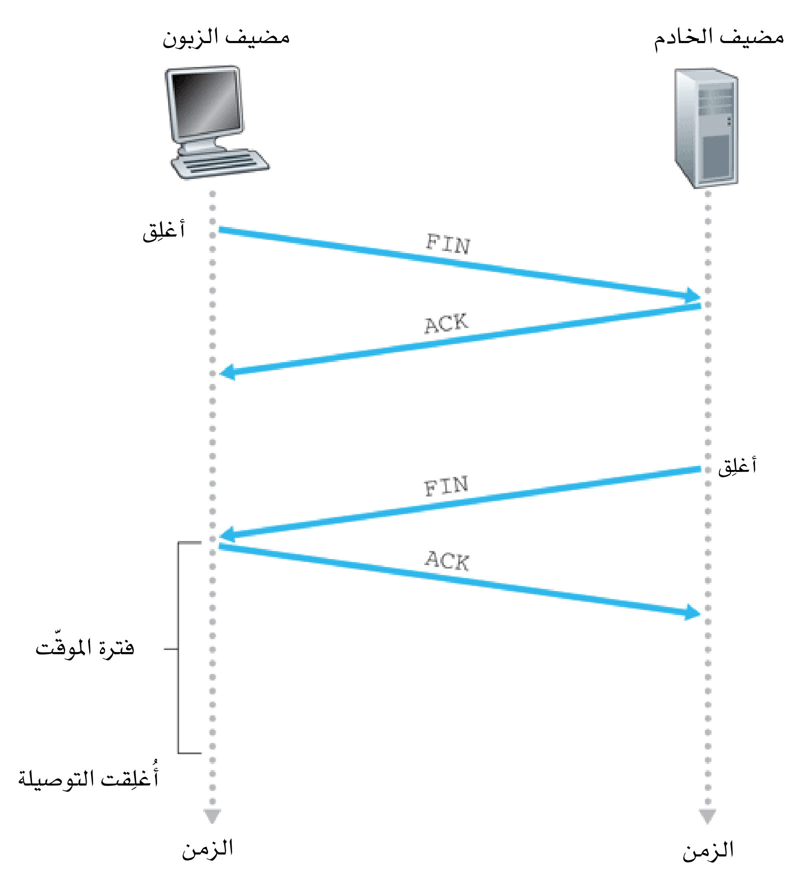
لماذا يتطلب الأمر مصافحة ثلاثية وليس فقط ثنائية؟). من الجدير بالملاحظة أن متسلق الصخور والمثبّت (الشخص الواقف على الأرض أسفل المتسلق والذي يضطلع بمهمة التعامل مع حبل أمان المتسلق) يستخدمان مصافحة ثلاثية تطابق مصافحة TCP لضمان أن كلا الجانبين جاهز قبل أن يبدأ المتسلق في الصعود.



الشكل 3-39 مصافحة بروتوكول TCP الثلاثية: تبادل القطع.

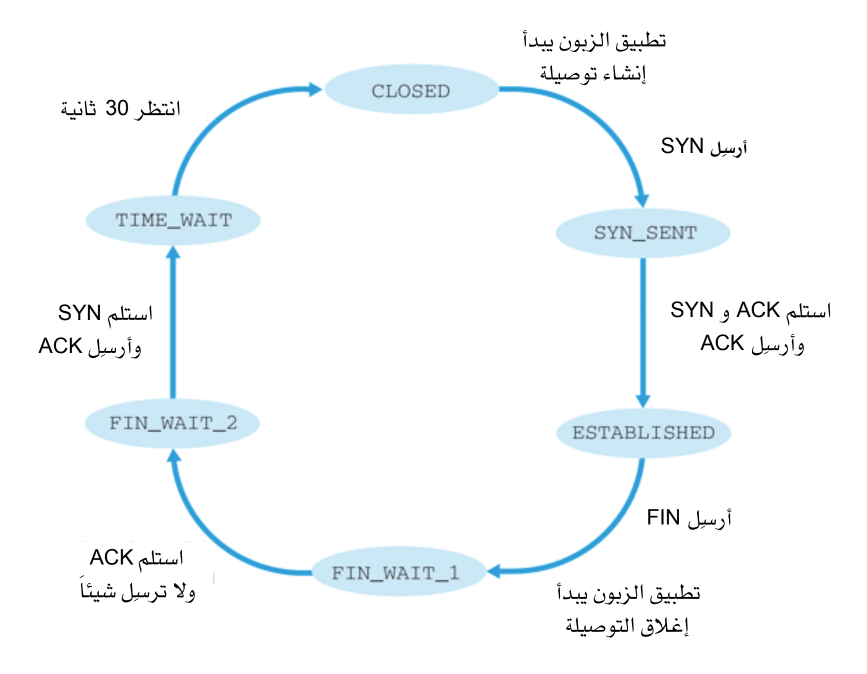
كل الأوقات الجميلة لا بد لها من نهاية، وهذا ينطبق أيضاً على توصيلات TCP. بوسع أيٍّ من العمليتين المشاركتين في توصيلة TCP إنهاء التوصيلة. وعند إنهاء توصيلة يتم إطلاق الموارد التي كانت تستخدمها تلك التوصيلة على كلٍّ من المضيفين (كالمخازن المؤقتة والمتغيرات) كي يتسنى استخدامها لتوصيلات أخرى. كمثال افترض أن الزبون قرّر إنهاء التوصيلة، كما هو مبين في الشكل 3-40. تُصدر عملية تطبيق الزبون أمراً بإنهاء التوصيلة. يؤدي ذلك إلى قيام بروتوكول

TCP على الزبون بإرسال قطعة TCP خاصة لها القيمة 1 للبت FIN في ترويسة القطعة إلى عملية الخادم (انظر الشكل 3-29). عندما يستلم الخادم هذه القطعة يرسل بدوره إلى الزبون إشعار استلام مقابل. بعد ذلك يرسل الخادم قطعة الإغلاق الخاصة به، والتي تكون قيمة البت FIN فيها 1 أيضاً. أخيراً يُرسل الزبون إشعاراً باستلام قطعة الإغلاق من الخادم. عند هذه النقطة، تكون كل موارد التوصيلة المنتهية على المضيفين قد تم إطلاقها وأصبحت متاحة لاستعمالات أخرى.



الشكل 3-40 إغلاق توصيلة TCP

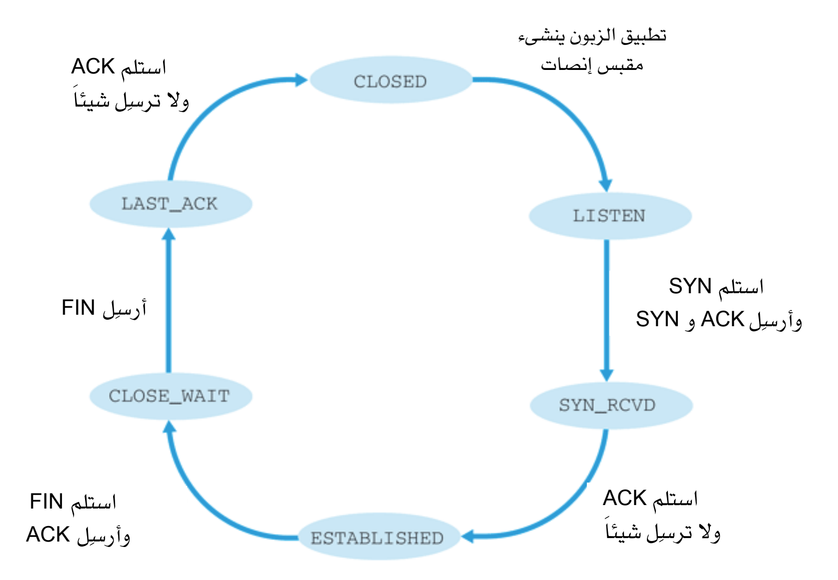
أثناء مدة توصيلة TCP يقوم بروتوكول TCP الذي يعمل على كلٍّ من المضيفين بالانتقال عبر عدة أوضاع مختلفة. يبين الشكل 3-41 تسلسلاً نمطياً للأوضاع التي يزورها بروتوكول TCP على الزبون. يبدأ زبون TCP في الوضع CLOSED (مغلق). يقوم التطبيق على جانب الزبون ببدء توصيلة TCP جديدة (بإنشاء كيان مقبس كما بيّنا في أمثلة جافا المذكورة في الفصل الثاني). يؤدي ذلك إلى قيام زبون TCP بإرسال قطعة SYN إلى خادم TCP. بعد إرسال القطعة SYN، يدخل زبون TCP وضع SYN_SENT (أُرسِلت قطعة SYN). ينتظر زبون TCP في ذلك الوضع وصول قطعة من خادم TCP تتضمّن إشعاراً باستلام قطعة الزبون السابقة وتحمل القيمة 1 للبت SYN. عند استلام تلك القطعة، يدخل زبون TCP وضع ESTABLISHED (التوصيلة شغّالة). في تلك الحالة يمكن لزبون TCP إرسال واستلام قطع TCP تتضمن بيانات (أي تولدها التطبيقات على مضيف الزبون).



الشكل 3-41 تسلسل نمطي لأوضاع TCP التي يزورها زبون TCP.

افترض أن تطبيق الزبون قرّر إنهاء التوصيلة – لاحظ أنه بوسع الخادم أيضاً أن يختار إنهاء التوصيلة – سيؤدي ذلك إلى إرسال زبون TCP قطعة TCP تحمل القيمة 1 للبت FIN، ومن ثم الدخول في وضع FIN_WAIT_1 منتظراً قطعة TCP من الخادم تتضمن إشعار استلام. عندما يتسلم الزبون تلك القطعة يدخل في وضع FIN_WAIT_2 منتظراً قطعة أخرى من الخادم تحمل القيمة 1 للبت FIN، وبعد استلام تلك القطعة يرسل زبون TCP إشعاراً باستلام قطعة الخادم ويدخل في وضع TIME_WAIT (الانتظار لفترة من الوقت) ليتيح الفرصة لبروتوكول TCP على الزبون لإرسال إشعار استلام نهائي في حالة فقد إشعار الاستلام السابق. ويعتمد وقت الانتظار الذي يقضيه الزبون في ذلك الوضع على إصدار TCP المستعمل، غير أن القيم المستخدمة عادةً هي 30 ثانية، ودقيقة، ودقيقتان. بعد الانتظار يتم إنهاء التوصيلة رسمياً وإطلاق كل الموارد التي كانت تستخدمها على جانب الزبون (بما في ذلك أرقام المنافذ).

يبين الشكل 3-42 تسلسلاً نمطياً للأوضاع التي يزورها بروتوكول TCP على جانب الخادم، على أساس أن الزبون يبدأ في إنهاء التوصيلة. الانتقالات بين الأوضاع المختلفة واضحة ولا تحتاج إلى شرح. وضّحنا في هذين المخططين للانتقال بين الأوضاع فقط الكيفية المعتادة لإنشاء وإنهاء توصيلة TCP، ولم نتعرّض لما يحدث في بعض السيناريوهات غير الطبيعية، على سبيل المثال عندما يرغب كلا الجانبين في إنشاء أو إنهاء توصيلة معاً في نفس الوقت. للمزيد من التفاصيل حول هذه النقطة وغيرها من القضايا المتقدّمة بخصوص بروتوكول TCP، ننصحك بالاطلاع على كتاب Stevens الشامل عن الموضوع [Stevens 1994].



الشكل 3-42 تسلسل نمطي لأوضاع TCP التي يزورها جانب الخادم من بروتوكول TCP.

افترضنا في مناقشتنا السابقة أن كلاًّ من الزبون والخادم مستعدان للاتصال، وبمعنى آخر أن الخادم يُنصت على المنفذ الذي يرسل إليه الزبون قطعة SYN. دعنا نتأمل ما يحدث عندما يستقبل مضيف قطعة TCP لها أرقام منافذ أو عنوان IP لا تتوافق مع أيٍّ من المقابس المستخدمة حالياً في المضيف. على سبيل المثال افترض أن مضيفاً يتسلم قطعة TCP من نوع SYN بمنفذ وجهة رقم 80، ولكن المضيف لا يقبل توصيلات على المنفذ 80 (أي أنه لا يقوم بتشغيل تطبيق خادم ويب على المنفذ 80). في تلك الحالة يرسل المضيف قطعة خاصّة Reset إلى المصدر تحمل القيمة 1 للبت RST (انظر الجزء 3-5-2)، ليخبر مضيف المصدر "ليس عندي مقبس لتلك القطعة، أرجو عدم إرسال القطعة مرة أخرى". أما عندما يتسلم مضيف قطعة بيانات UDP لا يتوافق منفذ الوجهة عليها مع أيٍّ من مقابس UDP الحالية على المضيف، يرسل المضيف وحدة بيانات ICMP خاصّة، كما سنبين في الفصل الرابع.

الآن وقد أصبحنا على دراية جيدة بإدارة توصيلات TCP، دعنا نزور مرة أخرى البرنامج nmap الخاص باستكشاف المنافذ لنرى عن قرب أكثر كيف

يعمل. لاستكشاف منفذ TCP بعينه، مثلاً منفذ 6789 على مضيف وجهة، يقوم برنامج nmap بإرسال قطعة SYN بمنفذ وجهة 6789 إلى ذلك المضيف. هناك ثلاث نتائج محتملة:

- يتسلم مضيف المصدر قطعة SYNACK من مضيف الوجهة، ولأن ذلك يعني وجود تطبيق يعمل على الوجهة بمنفذ TCP رقم 6789، فإن البرنامج nmap يعطي النتيجة "Open" (المنفذ مفتوح).
- يتسلم مضيف المصدر قطعة RST من مضيف الوجهة، وهذا يعني أن قطعة SYN قد وصلت لمضيف الوجهة إلا أنه ليس لديه تطبيق يعمل على منفذ TCP رقم 6789. بوسع المهاجم الآن على الأقل استنتاج أن القطع المُرسَلة إلى مضيف الوجهة بمنفذ 6789 لا يتم حجبها ببرنامج الحماية (firewall) على الطريق بين مضيفي المصدر والوجهة (سنناقش برامج الحماية في الفصل الثامن).
- لا يتسلم مضيف المصدر أي شيء، مما قد يعني أن القطعة SYN تم حجبها بأحد برامج الحماية على الطريق إلى مضيف الوجهة، ومن ثم لم تتمكن من الوصول إليه.

يعتبر برنامج nmap أداةً قويةً يمكن استخدامها في "فحص المبنى قبل السطو عليه"، ليس فقط لمنافذ TCP المفتوحة، ولكن أيضاً لمنافذ UDP المفتوحة، ولبرامج الحماية وإعداداتها، بل وحتى للإصدارات المستخدمة من التطبيقات وأنظمة التشغيل. يتم أغلب ذلك من خلال معالجة قطع TCP الخاصة بإدارة التوصيلات [Skoudis 2006]. إذا كنت الآن جالساً بالقرب من حاسب يعمل بنظام تشغيل لاينكس، فقد تريد تجربة البرنامج nmap بسرعة، فقط أدخل الأمر "nmap". أما إذا كنت تستخدم أحد أنظمة التشغيل الأخرى فبوسعك تنزيل البرنامج nmap من الموقع http://insecure.org/nmap.

بهذا نكون قد أنهينا مقدمتنا عن التحكم في الأخطاء وضبط التدفق في بروتوكول TCP. في الجزء 3-7 سنعود إلى بروتوكول TCP لنتناول بتفصيل أكثر كيفية التحكم في الازدحام. ولكن قبل ذلك دعنا نأخذ خطوة للوراء لكي نتفحص قضايا التحكم في الازدحام في سياق أوسع.

نبذة عن الأمن (Focus on Security)

هجوم فيضان SYN

رأينا من خلال مناقشتنا لمصافحة TCP الثلاثية أن الخادم يخصص ويضع القيم الأولية للمتغيرات والمخازن المؤقتة للتوصيلة كاستجابة لقطعة SYN التي يتسلمها. يقوم الخادم بعد ذلك بالرد بإرسال قطعة SYNACK، وينتظر وصول قطعة إشعار استلام ACK من الزبون، والتي تعتبر بمثابة الخطوة الثالثة والأخيرة في عملية المصافحة التي تسبق إنشاء توصيلة بالكامل. إذا لم يرسل الزبون إشعار الاستلام لإكمال الخطوة الثالثة من المصافحة الثلاثية فإنه في النهاية (غالباً بعد دقيقة أو أكثر) يقوم الخادم بإنهاء التوصيلة نصف المفتوحة، ويستردّ الموارد التي كان قد خصّصها لها.

يهيئ هذا الأسلوب لإدارة التوصيلة المسرح لهجوم شائع من نوع حجب الخدمة (DoS) يُعرف باسم فيضان SYN. في هذا الهجوم يرسل المهاجم عدداً كبيراً من قطع TCP SYN بدون إكمال الخطوة الثالثة من عملية المصافحة. يمكن مضاعفة أثر الهجوم بإرسال القطع SYN من مصادر متعدّدة، ومن ثم شن هجوم موزّع لحجب الخدمة (DDoS). بهذا الطوفان من قطع SYN يمكن بسرعة إنهاك موارد التوصيلات على الخادم حيث يتم تخصيصها لتوصيلات نصف مفتوحة (ولكنها لا تستخدم!). مع استنزاف مصادر الخادم بهذا الشكل يُحرَم زبائن شرعيون من الخدمة. ولقد كانت هجمات فيضان SYN تلك [CERT SYN 1996] بين هجمات حجب الخدمة الأولى التي تم توثيقها من قِبَل CERT [CERT 2007].

يمكن أن يكون هجوم فيضان SYN هجوماً مدمّراً فعلاً، لكن لحسن الحظ هناك دفاع فعّال ضده، يطلق عليه اسم كوكيز SYN (SYN cookies) والذي يُستخدم الآن في معظم أنظمة التشغيل الرئيسة [Skoudis 2006; Cisco SYN 2007; Bernstein 2007].

يعمل كوكيز SYN كالتّالي:

- عندما يتسلم الخادم قطعة SYN لا يعرف ما إذا كانت القطعة قادمة من مستخدم شرعي أو أنها جزء من هجوم فيضان SYN فإن الخادم لا ينشيء توصيلة TCP نصف مفتوحة لتلك القطعة، وإنما يقوم الخادم بتكوين رقم TCP تسلسلي أوّلي يكون دالة معقّدة (دالة تشفير الهاش hash function) في كلٍّ من عناوين IP للمصدر والوجهة وأرقام المنافذ لقطعة SYN، بالإضافة إلى رقم سري يعرفه الخادم فقط (ويستعمله لعدد كبير من التوصيلات). هذا الرقم التسلسلي الأوّلي والمحاك بعناية هو ما يسمّى بالكوكي. بعد

ذلك يرسل الخادم قطعة SYNACK بهذا الرقم التسلسلي الأوّلي الخاص. من المهم ملاحظة أن الخادم لا يتذكّر الكوكي ولا أي معلومات عن الأوضاع فيما يتعلق بقطعة SYN.

- إذا كان الزبون شرعياً، فإنه سيرد بقطعة إشعار استلام. عند استلام هذا الإشعار يقوم الخادم بالتحقق من أن الإشعار يقابل قطعة SYN التي أُرسلت في وقت سابق. كيف يقوم الخادم بذلك إذا كان لا يحتفظ في ذاكرته بأي شيء عن قطع SYN؟ كما قد تكون قد خمّنت، يتم ذلك باستخدام الكوكي. بالتحديد، إذا كان إشعار الاستلام شرعيا القيمة في حقل رقم إشعار الاستلام تزيد واحد عن الرقم التسلسلي لقطعة SYNACK التي أُرسِلت (انظر الشكل 3-39). يقوم الخادم باستخدام نفس الدالة مع نفس الحقول في قطعة إشعار الاستلام ونفس العدد السري. إذا كانت النتيجة تقل بواحد عن رقم إشعار الاستلام يستنتج الخادم أن إشعار الاستلام يناظر قطعة SYN سابقة ومن ثم فهو صحيح. يقوم الخادم بعد ذلك بإنشاء توصيلة مفتوحة بالكامل مع مقبس.
- من ناحية أخرى، إذا لم يرد الزبون بقطعة إشعار استلام، فإن قطعة SYN الأصلية لم تكن قد تسببت في أي ضرر عند الخادم، حيث إنه لم يقم بتخصيص أي موارد لها.

تتغلب طريقة كوكيز SYN بشكلٍ فعّال على خطر هجوم فيضان SYN. ومع ذلك هناك نوع من هجوم فيضان SYN يقوم فيه الزبون الماكر بإعادة قطعة إشعار استلام صحيحة لكل قطعة SYNACK يرسلها الخادم مما يؤدي إلى أن ينشىء الخادم توصيلات TCP مفتوحة بالكامل حتى في حالة استخدام نظام تشغيله لكوكيز SYN. كذلك إذا تم استخدام عشرات الآلاف من الزبائن – لكلٍ منهم عنوان مصدر IP مختلف – في الهجوم (أي هجوم موزّع لحجب الخدمة DDoS) يصبح من الصعب على الخادم التمييز بين المصادر الشرعية والخبيثة. وعليه فإن "هجوم المصافحة المكتملة" هذا يكون أكثر صعوبة في الدفاع ضده من هجوم فيضان SYN العادي.

3-6 مبادئ التحكم في الازدحام

في الفصول السابقة تناولنا كلاً من المبادئ العامة والآليات المحددة التي يستخدمها بروتوكول TCP لتوفير نقل موثوق للبيانات تحت ظروف فقد الرزم. وقد سبق أن ذكرنا أن فقد الرزم يرجع عادةً إلى فيض المخازن المؤقتة الموجودة في الموجّهات عند ازدحام الشبكة. ولذا فإن إعادة إرسال رزمة يعالج أحد أعراض ازدحام الشبكة وهو فقد قطعة بعينها من قطع طبقة النقل، ولكنه لا يعالج السبب الحقيقي لازدحام الشبكة — والذي يتلخص في محاولة الكثير من المصادر إرسال البيانات بمعدلات إرسال عالية جداً. لمعالجة سبب ازدحام الشبكة نحتاج إلى آليات للحد من قدرة الإرسال لدى المصادر لمواجهة ازدحام الشبكة.

في هذا الجزء سنتناول قضية التحكم في الازدحام في سياق عام لكي نفهم لماذا يُعتبر الازدحام شيئاً غير مرغوب فيه، وأثر هذا الازدحام على الأداء الذي توفره الشبكة لتطبيقات الطبقات الأعلى، وكذلك الطرق المختلفة التي يمكن اتباعها لتفادي حدوث ازدحام في الشبكة أو للتصرف إزاءه عند حدوثه. تعتبر هذه الدراسة العامة للسيطرة على الازدحام شيئاً مناسباً لأنها، كموضوع النقل الموثوق للبيانات، تحتل مرتبة متقدمة على قائمة أهم عشر مشاكل أساسية في مجال الشبكات. ثم بعد ذلك نختم هذا الجزء بمناقشة التحكم في الازدحام في خدمة معدّل البتات المتاح ((Available Bit Rate (ABR)) المستخدمة في شبكات نمط النقل غير المتزامن (ATM). أما الجزء التالي فيتضمن دراسة مفصلة لخوارزميات بروتوكول TCP للسيطرة على الازدحام.

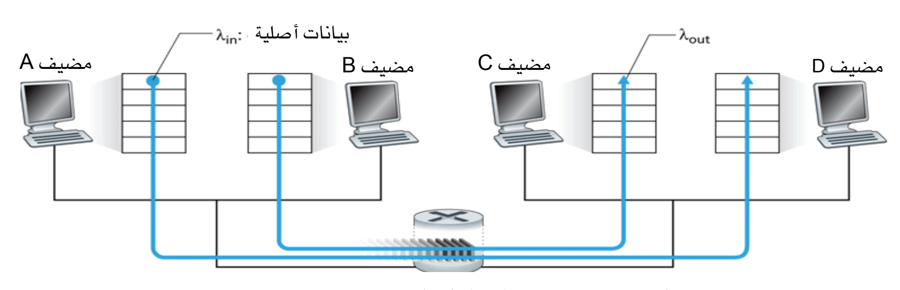
3-6-1 أسباب ازدحام الشبكة ومضاره

دعنا نبدأ دراستنا العامة للسيطرة على الازدحام بتناول ثلاثة سيناريوهات يحدث فيها ازدحام في الشبكة وتزداد تعقيداً بشكلٍ تدريجي. في كل حالة سنرى لماذا حدث الازدحام أساساً وماهو حجم الخسارة الناجمة عنه (من حيث الاستخدام غير الأمثل لموارد الشبكة والمستوى السيئ من الخدمة التي توفرها الشبكة

للأنظمة الطرفية). لن نركّز الآن على كيفية تفادي حدوث ازدحام أو كيفية التعامل معه عند حدوثه، ولكن سيكون التركيز على القضية الأسهل المتعلقة بفهم ما يحدث عندما تزيد المضيفات من معدلات إرسالها ومن ثم تزدحم الشبكة.

السيناريو الأول: مُرسِلان وموجّه بحيّز لانهائي للتخزين المؤقت

نبدأ بتناول ما قد يُعتبر أسهل سيناريو ازدحام ممكن: مضيفان A وB لكلٍ منهما توصيلة تشترك في وصلة واحدة بين المصدر والوجهة، كما هو موضح في الشكل 3-34.

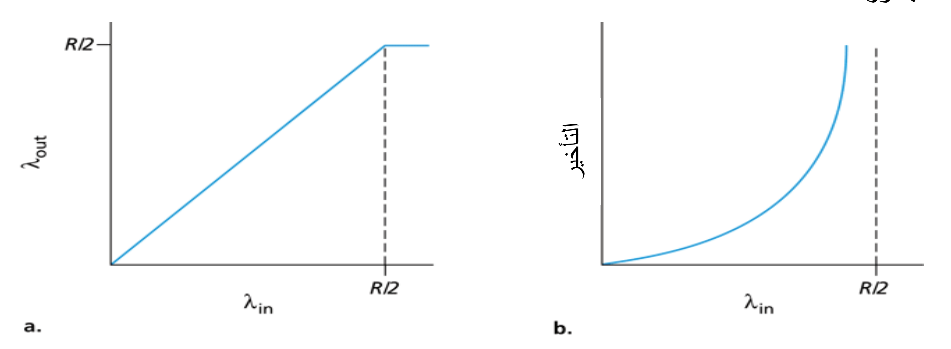


الشكل 3-43 سيناريو الازدحام رقم 1: توصيلتان تشتركان في وصلة واحدة بمخازن مؤقتة ذات سعة غير محدودة.

لنفترض أن التطبيق في المضيف A يرسل البيانات إلى التوصيلة (على سبيل المثال يدفع بالبيانات إلى بروتوكول طبقة النقل عبر المقبس) بمعدل إرسال متوسط مقداره λ_{in} بايت/ثانية. هذه البيانات كلها أصلية بمعنى أن كل وحدة بيانات ترسل إلى المقبس مرة واحدة فقط. افترض أن بروتوكول نقل البيانات التحتي بسيط، فالبيانات تغلّف ثم ترسل بدون أي آليات للتعافي من أخطاء البيانات (بإعادة الإرسال على سبيل المثال) أو لضبط التدفق أو للسيطرة على الازدحام. بإهمال الأعباء الإضافية الناجمة عن إضافة معلومات الترويسة على مستوى طبقة النقل والطبقات الأدنى، يكون معدل البيانات التي يرسلها المضيف A إلى الموجّه في هذا السيناريو

الأول λ_{in} بايت/ثانية. يعمل المضيف B بطريقة مماثلة، ولنفترض للتبسيط أنه أيضاً يرسل البيانات بمعدل λ_{in} بايت/ثانية. تمر الرزم من المضيفين A وB عبر الموجّه وعلى وصلة خارجة مشتركة لها سعة إرسال R بايت/ثانية. يحتوي الموجّه على مخزن مؤقت للرزم التي يتلقاها يُستخدم عندما يتجاوز معدل وصول الرزم سعة الإرسال الخاصة بالوصلة الخارجة. سنفترض أيضاً في هذا السيناريو الأول أن الموجّه لديه كمية لانهائية من حيّز التخزين.

تبيّن الرسوم البيانية في الشكل 3-44 أداء توصيلة المضيف A تحت ظروف هذا السيناريو الأول. يوضح الرسم البياني الأيسر الطاقة الإنتاجية لكل توصيلة (عدد البايتات التي تصل إلى المُستقبل كل ثانية) كدالة في معدل الإرسال على التوصيلة. لمعدلات الإرسال من 0 إلى $R/2$، تساوي الطاقة الإنتاجية عند المُستقبل معدل الإرسال عند المُرسِل – أي أن كل شيء يرسله المُرسِل يتم استلامه لدى المُستقبل بعد فترة تأخير محدودة. ولكن عندما يتجاوز معدل الإرسال القيمة $R/2$، تتوقف الطاقة الإنتاجية عندالقيمة $R/2$. ينشأ هذا الحدّ الأعلى للطاقة الإنتاجية بسبب اشتراك توصيلتي المضيفين في سعة إرسال الوصلة، أي أن الوصلة ببساطة لا تستطيع توصيل البايتات إلى المُستقبل بمعدل يتجاوز $R/2$. فمهما كان معدل الإرسال من كلٍّ من المضيفين A وB، فلن يحظى أي منهما أبداً بطاقة إنتاجية تتجاوز $R/2$.

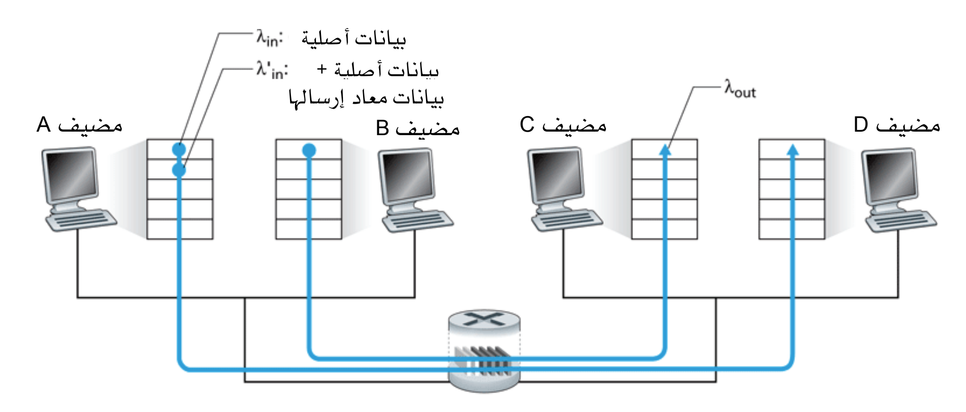


الشكل 3-44 سيناريو الازدحام رقم 1: الطاقة الإنتاجية والتأخير كدالة في معدل إرسال البيانات من الخادم.

قد يبدو تحقيق طاقة إنتاجية تعادل $R/2$ لكل توصيلة شيئاً جيداً في الواقع، لأن الوصلة تُستغَل بالكامل في توصيل الرزم إلى وجهاتها. غير أن الرسم البياني الأيمن في الشكل 3-44 يبين العواقب الوخيمة للتشغيل قرب سعة الإرسال القصوى لوصلة، فعند اقتراب معدل الإرسال على كلتا التوصيلتين من $R/2$ (بالزيادة من ناحية اليسار)، يزداد متوسط التأخير أكثر فأكثر. وعندما يتجاوز معدل الإرسال $R/2$، يصبح العدد المتوسط للرزم المنتظرة في الصف الخاص بالوصلة الخارجة بالموجّه غير محدود، ومن ثم يكون متوسط وقت التأخير بين المصدر والوجهة لانهائياً (على افتراض أن التوصيلات تبقى تعمل على ما يرام عند معدلات الإرسال هذه لفترات لانهائية وأنه تتوافر كمية لانهائية من حيّز التخزين المؤقت). وهكذا فبينما يكون التشغيل عند طاقة إنتاجية كلية قريبة من R أمراً مثالياً من منظور الطاقة الإنتاجية، إلا أنه يكون غير مرغوب فيه من حيث وقت التأخير المتزايد. حتى في هذا السيناريو المثالي للغاية، وجدنا عيباً واضحاً لازدحام الشبكة — تأخيرات انتظار كبيرة تعاني منها الرزم عندما تقترب معدلات وصول بيانات الرزم من سعة الإرسال للوصلة.

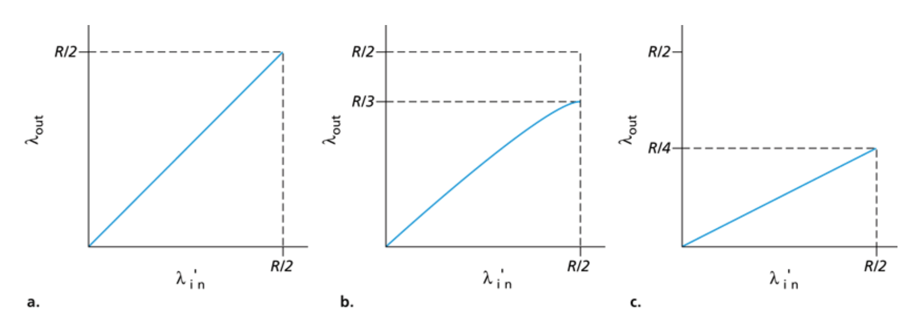
السيناريو الثاني: مُرسِلان وموجّه بحيّز محدود للتخزين المؤقت

دعنا الآن نُدخِل بعض التعديلات على السيناريو الأول من ناحيتين (انظر الشكل 3-45). أولاً: افترض أن حيّز التخزين المؤقت محدود. كنتيجة لهذا الافتراض الواقعي ستُفقَد الرزم التي تصل إلى الموجّه فتجد حيّز التخزين ممتلئاً عن آخره. ثانياً: افترض أن كل توصيلة توفر نقلاً موثوقاً للبيانات، فإذا فقدت في الموجّه رزمة تتضمن قطعة من قطع طبقة النقل، فإن المُرسِل سيعيد إرسالها في النهاية. ولأن الرزم يمكن أن يعاد إرسالها الآن، علينا أن نكون أكثر حذراً في استخدامنا لمصطلح معدل الإرسال. وبتحديد أكثر دعنا نرمز مرة أخرى للمعدل الذي يرسل به التطبيق بياناته الأصلية إلى المقبس بالرمز λ_{in} بايت/ثانية. أما المعدل الذي ترسل به طبقة النقل البيانات (قطع تحتوي بيانات أصلية وقطع معاد إرسالها) إلى الشبكة فسنرمز له بالرمز λ'_{in} بايت/ثانية. أحياناً يطلق على λ'_{in} الحمل المقدم إلى الشبكة (offered load).



الشكل 3-45 سيناريو الازدحام رقم 2: مضيفان (بإعادة إرسال) وموجّه بمخازن مؤقتة ذات سعة محدودة.

سيعتمد الأداء الذي يمكن تحقيقه تحت ظروف السيناريو الثاني بشكلٍ قوي الآن على كيفية القيام بإعادة الإرسال. أولاً خذ في الاعتبار الحالة غير الواقعية التي يتمكن فيها المضيف A بشكلٍ أو بآخر (بطريقة سحرية!) أن يحدد ما إذا كان هناك مكان خالٍ أم لا في المخزن المؤقت على الموجّه، ومن ثم يرسل رزمة فقط إذا توفر مكان للتخزين هناك. في هذه الحالة لا يحدث فقد للرزم ومن ثم لا يكون هناك داعٍ لإعادة الإرسال، وعليه يكون λ_{in} مساوياً لـ λ'_{in}، وتكون الطاقة الإنتاجية للتوصيلة λ_{in}. يبين الشكل 3-46 (a) هذه الحالة. من منظور الطاقة الإنتاجية يُعدّ الأداء فيها مثالياً، فكل شيء يتم إرساله يتم استلامه. لاحظ أن متوسط معدل الإرسال من المضيف لا يمكن أن يتجاوز $R/2$ في هذا السيناريو حيث افترضنا أن فقد الرزم لا يمكن أن يحدث أبداً.



الشكل 3-46 سيناريو الازدحام رقم 2: الأداء في حالة استخدام مخازن مؤقتة ذات سعة محدودة.

لنأخذ بعين الاعتبار بعد ذلك الحالة الأكثر واقعية بعض الشيء والتي يعيد فيها المُرسِل الإرسال فقط عندما يعرف بالتأكيد أن رزمة قد فُقدت (مرةً أخرى هذه الفرضية مسرفة في الخيال بعض الشيء، ولكن من الممكن أن يضبط المضيف فترة الموقّت عند قيمة كبيرة بما فيه الكفاية بحيث يتأكد عملياً من أن الرزمة التي لم يصل إشعار باستلامها تكون فعلاً قد فُقدت). في هذه الحالة يمكن أن يبدو الأداء كما هو مبين في الشكل 3-46 (b). لإدراك ما يحدث هنا تأمل الحالة التي يكون فيها الحمل المقدم للشبكة λ'_{in} (معدل إرسال البيانات الأصلية بالإضافة إلى البيانات المعاد إرسالها) يساوي $R/2$. حسب الشكل 3-46 (b)، عند هذه القيمة للحمل المقدم للشبكة، يكون معدل توصيل البيانات إلى تطبيق المُستقبِل $R/3$. وعليه، فإنه من كل $0.5R$ بايت/ثانية أُرسلت، هناك 0.333R بايت/ثانية (في المتوسط) بيانات أصلية و$0.166R$ بايت/ثانية (في المتوسط) بيانات يعاد إرسالها. نرى الآن كلفة أخرى لاستخدام شبكة مزدحمة - على المُرسِل أن يعيد الإرسال لكي يعوّض فقد الرزم بسبب فيض المخزن المؤقت بالموجّه.

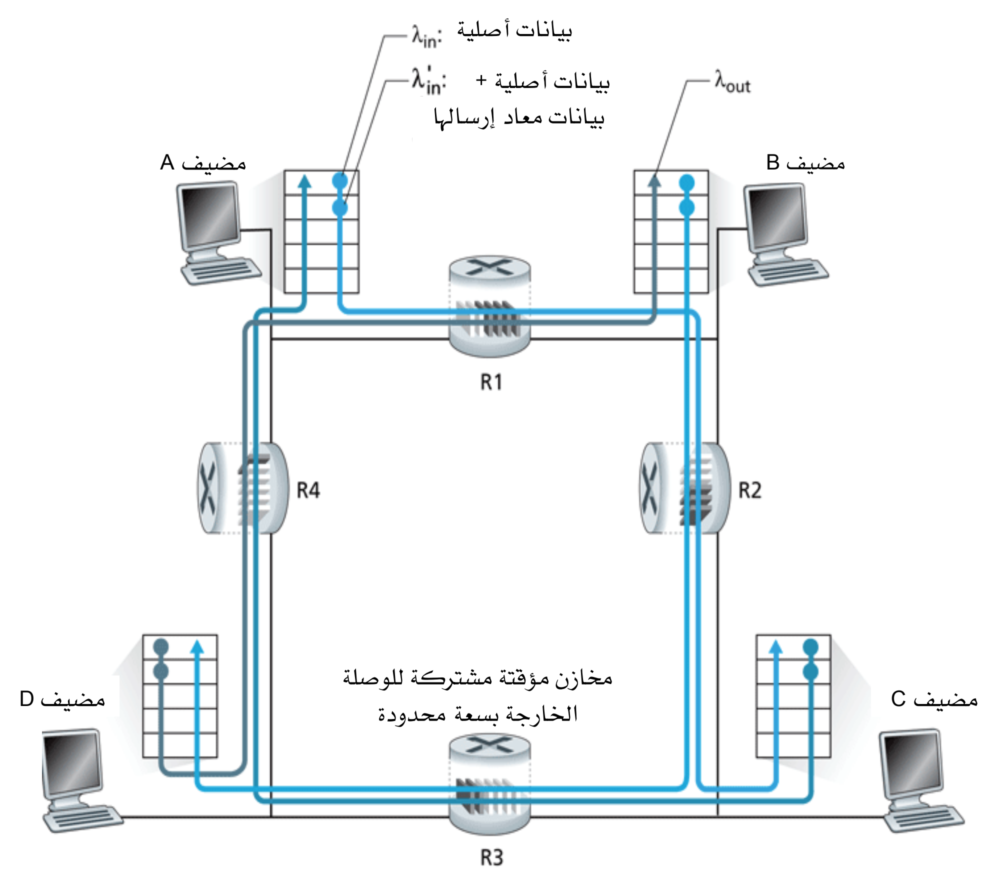
دعنا أخيراً نأخذ بعين الاعتبار الحالة التي قد تنقضي فيها فترة الموقّت عند المُرسِل قبل الأوان بحيث يعيد إرسال رزمة تأخّرت في صف الانتظار على الموجّه ولكنها لم تفقد بعد. في هذه الحالة قد تصل كلٌّ من رزمة البيانات الأصلية ورزمة إعادة الإرسال إلى المُستقبِل. وبالطبع فإن المُستقبِل يحتاج لنسخة واحدة فقط من تلك الرزمة وسيهمل رزمة إعادة الإرسال، وبذلك يعتبر الجهد الذي بذله الموجّه في إعادة

إرسال نسخة من الرزمة الأصلية جهداً مهدراً، لأن المُستقبل سيكون قد تسلم النسخة الأصلية من تلك الرزمة. وبدلاً من ذلك كان يمكن للموجّه استغلال سعة إرسال الوصلة بشكلٍ أفضل لإرسال رزمة مختلفة. هنا إذن نلاحظ كلفة أخرى لاستخدام شبكة مزدحمة - عمليات إعادة الإرسال بدون داعٍ من قِبَل المُرسِل بسبب التأخيرات الكبيرة، والتي قد تتسبب في جعل الموجّه يستخدم سعة الإرسال لوصلته لتوجيه نسخ غير مطلوبة من الرزم. يبين الشكل 3-46 (c) الطاقة الإنتاجية مقابل الحمل المقدّم للشبكة على افتراض أن كل رزمة ترسل (في المتوسط) مرتين بواسطة الموجّه. ونظراً لأن كل رزمة ترسل مرتين، فإن الطاقة الإنتاجية سيكون لها قيمة تقاربية (asymptotic value) مقدارها $R/4$ عندما يقترب الحمل المقدم للشبكة من $R/2$.

السيناريو الثالث: أربعة مُرسِلين، وموجّهين بحيّز محدود للتخزين المؤقت، ومسارات بعدة وصلات

في هذا السيناريو الأخير يقوم أربعة مضيفات بإرسال الرزم، كلٌّ على مسارات متطابقة تضم وصلتين، كما هو موضح في الشكل 3-47. نفترض مرة أخرى أن كل مضيف يستخدم آلية انقضاء فترة موقّت إعادة الإرسال لتوفير خدمة نقل موثوق للبيانات، وأن كل المضيفات لها نفس قيمة معدل إرسال البيانات الأصلية λ_{in}، وأن كل وصلات الموجّهات لها سعة إرسال R بايت/ثانية.

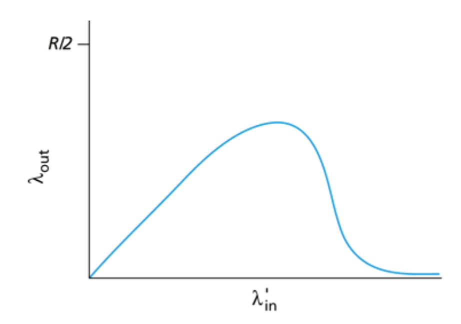
لنأخذ في الاعتبار التوصيلة من المضيف A إلى المضيف C مروراً بالموجّهين R1 وR2. تشترك التوصيلة A↔C في الموجّه R1 مع التوصيلة D↔B وفي الموجّه R2 مع التوصيلة B↔D. للقيم الصغيرة جداً لـ λ_{in} يكون فيض المخازن المؤقتة نادراً (كما في سيناريوهات الازدحام الأول والثاني)، وتكون الطاقة الإنتاجية مساوية تقريباً للحمل المقدم للشبكة. للقيم الأكبر قليلاً لـ λ_{in} تكون الطاقة الإنتاجية المناظرة أكبر أيضاً، حيث يتم إرسال بيانات أصلية أكثر إلى الشبكة وتوصيلها إلى الوجهة، ويبقى الفيض نادراً. وهكذا فإنه لقيم λ_{in} الصغيرة، تؤدي الزيادة في λ_{in} إلى زيادة في λ_{out}.



الشكل 3-47 سيناريو الازدحام رقم 3: أربعة مضيفات مُرسِلة، وموجّهات بمخازن مؤقتة ذات سعة محدودة، ومسارات متعددة الوصلات.

بعد أن تناولنا حالة حركة مرور البيانات المنخفضة جداً، دعنا ندرس الآن الحالة التي تكون فيها λ_{in} (وبالتالي λ'_{in}) كبيرة جداً. خذ في الاعتبار الموجّه R2. إن حركة مرور بيانات A↔C الواصلة إلى الموجّه R2 (والتي تصل إليه بعد توجيهها من R1) يمكن أن يكون لها معدل وصول عند R2 كحد أقصى R، أي سعة الإرسال على الوصلة من R1 إلى R2، بغض النظرعن قيمة λ_{in}. إذا كانت λ'_{in} كبيرة جداً لكل التوصيلات (بما في ذلك التوصيلة B↔D)، فإن معدل وصول حركة المرور على B↔D عند R2 يمكن أن يكون أكبر بكثير من معدل وصول حركة المرور

على A↔C. ونظراً لأنه على حركتي المرور A↔C وB↔D أن تتنافسا على السعة المحدودة لحيّز التخزين المؤقت في الموجّه R2، فإن كمية حركة المرور A↔C التي تعبر R2 بنجاح (أي دون أن تُفقد هناك بسبب فيض المخزن المؤقت) تصبح أقل فأقل عندما يصبح الحمل المقدّم للشبكة من B↔D أكبر فأكبر. في النهاية عندما يقترب الحمل المقدم للشبكة من اللانهاية، فإن الفراغ في المخزن المؤقت في R2 سيُملأ فوراً برزمة من B↔D، وبذا تضمحل الطاقة الإنتاجية للتوصيلة A↔C عند الموجّه R2 لتصل إلى صفر. وهذا يعني بدوره أن الطاقة الإنتاجية للتوصيلة A↔C من طرف إلى طرف تصل إلى صفر في النهاية في حالة الازدحام الشديد. تؤدي تلك الاعتبارات إلى العلاقة المبينة في الشكل 3-48 والتي توضح الموازنة بين الحمل المقدم للشبكة في مقابل الطاقة الإنتاجية.



الشكل 3-48 سيناريو الازدحام رقم 3: الأداء في حالة موجّهات بمخازن مؤقتة ذات سعة محدودة ومسارات متعددة الوصلات.

يتضح السبب في النقص النهائي في الطاقة الإنتاجية عند زيادة الحمل المقدم للشبكة عندما نأخذ بعين الاعتبار كمية الشغل المهدر الذي تبذله الشبكة. في سيناريو حركة المرور العالية الذي تناولناه من قبل، كلما سقطت رزمة في موجّه الوصلة الثانية، فإن الشغل الذي بذله موجّه الوصلة الأولى في توجيه الرزمة إلى موجّه الوصلة الثانية يكون قد ضاع هباءً. كان من الممكن أن تكون الشبكة على نفس الحالة من الجودة (أو بتعبير أدق على نفس الحالة من السوء!) إذا كان الموجّه الأول ببساطة قد أهمل تلك الرزمة وبقى خاملاً. وبتحديد أكثر فإن قدرة

الإرسال التي استُخدمت في المسار الأول لإرسال تلك الرزمة إلى الموجّه الثاني كان يمكن استغلالها بشكلٍ أكثر فائدة بكثير لإرسال رزمة مختلفة (على سبيل المثال عند اختياره لرزمة يرسلها، قد يكون من الأفضل للموجّه إعطاء الأولوية لرزم قد قطعت شوطاً في رحلتها وعبرت عدة موجّهات في الطريق إلى وجهتها النهائية). ومن هنا يتضح لنا كلفة أخرى لإسقاط الرزم بسبب الازدحام – فعند إسقاط رزمة على مسارٍ ما، فإن سعة الإرسال التي استعملت في كل الوصلات السابقة لإيصال تلك الرزمة إلى تلك النقطة من الشبكة التي أُسقطت منها الرزمة تكون قد أُهدرت للأسف.

3-6-2 طرق التحكم في الازدحام

سنتناول في الجزء 3-7 بتفصيلٍ أكثر الطرق التي يتبعها بروتوكول TCP للتحكم في الازدحام. سنتعرف هنا فقط على أسلوبين عامّين للتحكم في الازدحام يُستخدمان في الواقع العملي، ونناقش بِنَى معمارية وبروتوكولات معينة لشبكات تتجسد فيها تلك الأساليب.

وعلى المستوى الأوسع يمكننا التمييز بين أساليب التحكم في الازدحام تبعاً لما إذا كانت طبقة الشبكة تقدم أي مساعدة واضحة لطبقة النقل لأغراض التحكم في الازدحام:

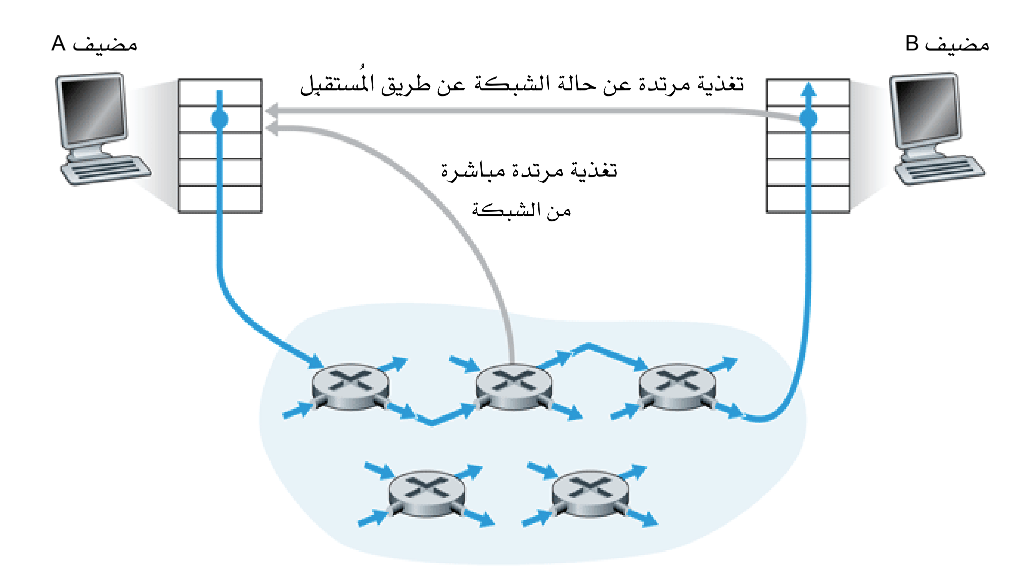
- أسلوب التحكم في الازدحام من طرف إلى طرف: في أساليب التحكم في الازدحام من طرف إلى طرف، لا تقدم طبقة الشبكة أي دعم محدد لطبقة النقل لأغراض التحكم في الازدحام، حتى اكتشاف وجود الازدحام في الشبكة يقع على عاتق الأنظمة الطرفية بناءً على ملاحظتها فقط لسلوك الشبكة (على سبيل المثال ملاحظة فقد الرزم والتأخير). سنرى في الجزء 3-7 أن بروتوكول TCP يجب أن يعتمد أسلوب من طرف إلى طرف هذا في التحكم في الازدحام، حيث لا يوفر بروتوكول طبقة الشبكة IP أي تغذية مرتدة (feedback) للأنظمة الطرفية بخصوص ازدحام الشبكة. إن فقد قطع بيانات TCP (والتي يدل عليها انقضاء فترة الموقّت أو وصول ثلاث إشعارات

استلام مكرّرة) ستؤخذ كمؤشر على ازدحام الشبكة، وبناءً عليه سيخفّض بروتوكول TCP من مقاس نافذته وفقاً لذلك. سنرى أيضاً اقتراحاً أحدث للتحكم في الازدحام في بروتوكول TCP يستخدم القيم المتزايدة لتأخير رحلة الذهاب والإياب كمؤشر على زيادة ازدحام الشبكة.

- أسلوب التحكم في الازدحام بمساعدة من الشبكة: في أسلوب التحكم في الازدحام بمساعدة من الشبكة، تقوم مكوّنات طبقة الشبكة (أي الموجّهات) بتقديم تغذية مرتدة صريحة إلى المُرسِل بخصوص حالة الازدحام في الشبكة. قد تكون تلك التغذية المرتدة بسيطة، مثلاً بت واحدة تشير إلى الازدحام في وصلة، وقد اتبع هذا الأسلوب في الشبكات الأولى كشبكات IBM SNA [Schwartz 1982] وDECnet [Jain 1989; Ramakrishnan 1990]، كما اقتُرح نفس الأسلوب مؤخراً لشبكات TCP/IP [Floyd TCP 1994; RFC 3168]، ويُستخدم أيضاً في شبكات ATM ضمن أسلوب ABR (معدل البتات المتاح) للتحكم في الازدحام والذي سنناقشه لاحقاً. هناك طرق أكثر تعقيداً لتوفير المزيد من تلك التغذية المرتدة، فمثلاً يسمح أحد أنواع بروتوكول ABR للتحكم في الازدحام على شبكات ATM – والذي سندرسه بعد قليل – للموجّه على الشبكة بإخبار المُرسِل صراحةً بمعدّل الإرسال الذي يمكن للموجّه دعمه على وصلة خارجة. وكذلك يوفر بروتوكول XCP (أي بروتوكول التحكم الصريح في الازدحام eXplicit Congestion control Protocol) تغذيةً مرتدة يقوم الموجّه بحسابها وإرسالها إلى كل مصدر ضمن ترويسة الرزمة، لتخبر المصدر بالكيفية التي يمكنه بها أن يزيد أو ينقص من معدل إرساله [Katabi 2002].

في أسلوب التحكم في الازدحام بمساعدة من الشبكة، غالباً ما تُرسَل المعلومات المتعلقة بالازدحام على شكل تغذية مرتدة إلى المُرسِل بإحدى طريقتين، كما هو موضحٌ في الشكل 3-49. قد تُرسَل التغذية المرتدة مباشرةً من موجّه على الشبكة إلى المُرسِل. عادةً ما يأخذ هذا الإخطار شكل رزمة خنق (choke packet) (لتقول على لسان الموجّه "أنا مزدحم!"). أما الشكل الثاني من الإخطارات فيحدث

عندما يقوم الموجّه بتعليم أو تحديث حقل في رزمة تعبره أثناء تدفّقها من المُرسِل إلى المُستقبِل للدلالة على حالة الازدحام. عند استلام حزمة معلّمة من موجّه يقوم المُستقبِل بإشعار المُرسِل بالازدحام. لاحظ أن هذا الشكل الأخير من الإخطارات يأخذ على الأقل وقتاً كاملاً لرحلة ذهاب وإياب.



الشكل 3-49 مساران للتغذية المرتدة بمعلومات عن ازدحام الشبكة.

3-6-3 مثال لأسلوب التحكم في الازدحام بمساعدة من الشبكة: أسلوب معدّل البتات المتاح (ABR) للسيطرة على الازدحام في شبكات ATM

سنختم هذا الجزء بدراسة حالة لخوارزمية للتحكم في الازدحام في بروتوكول ABR (معدّل البتات المتاح) المستخدم على شبكات نمط النقل غير المتزامن (Asynchronous Transfer Mode (ATM)) والذي يعتمد على أسلوب التحكم في الازدحام بمساعدة من الشبكة. نؤكد أن هدفنا هنا ليس وصف السمات المعمارية لشبكات ATM بالتفصيل، ولكن فقط توضيح بروتوكول يسلك مسلكاً مختلفاً بدرجة كبيرة عن بروتوكول TCP على الإنترنت في أسلوب

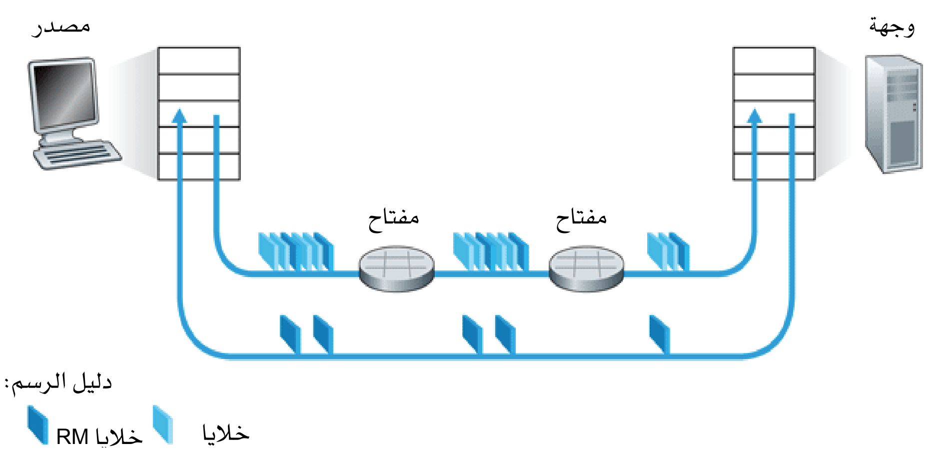
التحكم في الازدحام. سوف نقتصر فيما يلي على استعراض عدد من السمات القليلة لعمارة شبكات ATM والتي نحتاجها لفهم أسلوب ABR للتحكم في الازدحام.

من حيث المبدأ، تتبع شبكات ATM أسلوب الدائرة الافتراضية (VC) لتحويل رزم البيانات. تذكّر من مناقشتنا في الفصل الأول أن هذا يعني أن كل محوّل على المسار من المصدر إلى الوجهة يحتفظ لديه بحالة الدائرة الافتراضية من المصدر إلى الوجهة. هذه المعلومات عن حالة كل دائرة افتراضية تمكّن المحوّل من تتبع سلوك مصادر البيانات كلٍّ على حدة (مثلاً تتبع المعدلات المتوسطة للإرسال لديها) واتخاذ إجراءات معينة على المصدر للتحكم في الازدحام (كإخطار المُرسِل صراحةً بضرورة تخفيض معدل إرساله عندما يصبح المحوّل مزدحماً). إن الاحتفاظ بمعلومات الحالة لكل دائرة افتراضية على محوّلات الشبكة بهذا الشكل يجعل التحكم في الازدحام بمساعدة الشبكة مناسباً في شبكات ATM.

تم تصميم بروتوكول ABR لتوفير خدمة مرنة لنقل البيانات بطريقة تشبه بروتوكول TCP، فعندما تكون الشبكة غير مزدحمة، ينبغي أن تكون خدمة ABR قادرة على استغلال الحيّز الترددي المتاح. وعندما تكون الشبكة مزدحمة، ينبغي أن تُحِد الخدمة من معدل إرسالها إلى معدل إرسال أدنى محدّد مسبقاً. يمكنك الاطلاع على دراسة تدريبية عن طريقة التحكم في الازدحام وإدارة حركة مرور البيانات ببروتوكول ABR في [Jain 1996].

يبين الشكل 3-50 إطار التحكم في الازدحام في بروتوكول ABR. سنستخدم في مناقشتنا التالية مصطلحات ATM (على سبيل المثال سنستخدم "محوّل" بدلاً من "موجّه"، و"خلية" بدلاً من "رزمة"). في خدمة ABR، ترسل خلايا البيانات من مصدر إلى وجهة عبر سلسلة من المحوّلات بينهما، تتخلل خلايا البيانات تلك خلايا إدارة موارد الشبكة ((Resource Management (RM)، والتي يمكن استخدامها لنقل المعلومات المتعلقة بالازدحام بين المضيفات والمحوّلات. عندما تصل خلية RM إلى وجهة، يتم تدويرها وإعادتها إلى المُرسِل (ربما بعد تعديل محتوياتها

بواسطة الوجهة). يمكن أيضاً لمحوّل إنشاء خلية RM جديدة بنفسه وإرسالها مباشرةً إلى المصدر. وعليه يمكن استخدام خلايا RM لتوفير تغذية مرتدة من الشبكة مباشرةً إلى المُرسِل أو بطريقة غير مباشرة عن طريق المُستقبل، كما هو مبين في الشكل 3-50.



الشكل 3-50 آلية التحكم في الازدحام مع خدمة ABR على شبكات ATM.

يعتمد بروتوكول ABR في التحكم في الازدحام على أسلوب مبني على معدل إرسال البتات، بمعنى أن المُرسِل يحسب بشكلٍ محدد أقصى معدل إرسال يمكن أن يستخدمه وينظّم نفسه وفقا لذلك. يوفر بروتوكول ABR ثلاث آليات لنقل المعلومات المتعلقة بالازدحام من المحوّلات إلى المُستقبل:

- البت EFCI: تحتوي كل خلية بيانات على بت للبيان الصريح للازدحام الأمامي (أي في اتجاه الوجهة) (Explicit Forward Congestion Indication (EFCI)). يمكن لمحوّل في شبكة مزدحمة وضع البت EFCI بالقيمة 1 لإخطار مضيف الوجهة بوجود ازدحام. يجب على مضيف الوجهة فحص البت EFCI على كل خلايا البيانات التي يتسلمها. عند وصول خلية RM إلى المضيف، إذا كانت البت EFCI في آخر خلية بيانات تم استلامها

تساوى 1، فإن مضيف الوجهة يضع 1 في بت CI (إشارة الازدحام Congestion Indication) ضمن خلية RM ويُرسلها في الاتجاه الآخر إلى المُرسِل. وهكذا فباستخدام البت EFCI في خلايا البيانات والبت CI في خلايا RM يمكن إخطار المُرسِل بحالة الازدحام عند محوّل بالشبكة.

- زوج البتات CI وNI: كما ذكرنا سابقاً تتخلل خلايا البيانات خلايا RM. إن نسبة خلايا RM الموزّعة وسط خلايا البيانات هي متغير قابل للضبط، يبدأ بقيمة أصلية مقدرها خلية RM واحدة لكل 32 خلية بيانات. تتضمن خلايا RM تلك زوجاً من البتات يمكن تغييرهما بواسطة محوّل على شبكة مزدحمة هما البت CI (إشارة الازدحام Congestion Indication) والبت NI (عدم الزيادة No Increase). بتحديد أكثر يمكن للمحوّل وضع القيمة 1 للبت NI في خلية RM تعبره في حالة ازدحام معتدل للشبكة، كما يمكنه وضع القيمة 1 للبت CI عند ظروف الازدحام الشديد. عندما يتسلم مضيف الوجهة خلية RM يقوم بإعادة تلك الخلية إلى المُرسِل بنفس قيم زوج البتات CI وNI التي استلمها (إلا أن البت CI يمكن أن تتغير إلى 1 في مضيف الوجهة كنتيجة لآلية البت EFCI المذكورة أعلاه).
- معدل الإرسال المحدد (Explicit Rate (ER)): تتضمن كل خلية RM أيضاً حقل معدل الإرسال المحدد (ER) والذي يتألف من بايتين. يمكن لمحوّل مزدحم تقليل قيمة الحقل ER في خلية RM تعبره. وبهذه الطريقة فإن حقل ER سيتم ضبطه بحيث يمثل الحد الأدنى لمعدل الإرسال الذي يمكن دعمه على كل المحوّلات الموجودة على المسار من المصدر إلى الوجهة.

يقوم مضيف المصدر في بروتوكول ATM بضبط معدل الإرسال الذي يمكنه استخدامه لبث الخلايا بناءً على قيم زوج البتات CI وNI والحقل ER في خلية RM الراجعة إليه من مضيف الوجهة. إن القواعد التي تحكم عملية ضبط معدل الإرسال

تلك معقدة ومجهِدة نوعًا ما، وعلى القارئ المهتم بالمزيد من التفاصيل مراجعة [Jain 1996].

7-3 التحكم في الازدحام في بروتوكول TCP

نعود في هذا الجزء إلى دراستنا لبروتوكول TCP. كما عرفنا في الجزء 3-5، يوفر TCP خدمة نقل موثوقة للبيانات بين عمليتين يتم تنفيذهما على مضيفين مختلفين. من المكونات الرئيسة الأخرى لبروتوكول TCP آلية التحكم في الازدحام. كما بينّا في الجزء السابق يجب أن يستخدم TCP أسلوب التحكم في الازدحام من طرف إلى طرف بدلاً من التحكم بمساعدة من الشبكة، نظراً لأن طبقة الشبكة (IP) لا توفر أي تغذية مرتدة إلى الأنظمة الطرفية بخصوص ازدحام الشبكة.

يتلخص الأسلوب الذي يتبعه بروتوكول TCP في جعل كل مُرسِل يُحِد من معدل إرساله للبيانات كدالة في الازدحام المحسوس للشبكة. فإذا أحس مُرسِل TCP أن الازدحام قليل على المسار بينه وبين الوجهة، فإن المُرسِل يزيد من معدل إرساله للبيانات. أما إذا شعر المُرسِل بأن هناك ازدحاماً على المسار، فإنه يخفض من معدل إرساله. غير أن هذا الأسلوب يطرح ثلاثة أسئلة. أولاً: كيف يمكن لمُرسِل TCP الحد من المعدل الذي يرسل به البيانات إلى التوصيلة؟ ثانياً: كيف يدرك المُرسِل أن هناك ازدحاماً في الشبكة على المسار بينه وبين الوجهة؟ وثالثاً: ما هي الخوارزمية التي يجب على المُرسِل استعمالها لتغيير معدل إرساله كدالة في الازدحام المحسوس في الشبكة من طرف إلى طرف؟ سنتناول هذه القضايا الثلاث في سياق الإصدار رينو Reno لبروتوكول TCP والذي يُستخدم في معظم أنظمة التشغيل الحديثة [Padhye 2001]. وللحفاظ على الطابع العملي للمناقشة سنفترض أن مُرسِل TCP يقوم بإرسال ملف كبير.

لنتناول أولاً كيف يُحد مُرسِل TCP من معدل إرساله للبيانات على التوصيلة. رأينا في الجزء 3-5 أن كلاً من جانبي توصيلة TCP يتكون من مخزن مؤقت

للاستقبال، ومخزن مؤقت للإرسال، وعدة متغيرات (RcvWindow،LastByteRead ، وهكذا). تتطلب آلية TCP للتحكم في الازدحام أن يتابع كل جانب من جانبي التوصيلة قيمة متغير إضافي يطلق عليه نافذة الازدحام (congestion window) والذي يرمز له بالرمز CongWin. يفرض المتغير CongWin قيداً على المعدل الذي يمكن لمُرسِل TCP استخدامه لإرسال البيانات إلى الشبكة. بالتحديد لا يُسمح لكمية البيانات التي أُرسلت ولم يتم الإشعار باستلامها عند مُرسِل أن تتجاوز القيمة الأصغر لكلٍّ من CongWin وRcvWindow، أي:

$$\text{LastByteSent} - \text{LastByteAcked} \leq \min\{\text{CongWin}, \text{RcvWindow}\}$$

لكي نركّز على التحكم في الازدحام (وليس ضبط التدفق)، دعنا نفترض من الآن فصاعداً أن مُستقبل TCP له مخزن استلام مؤقت كبير جداً بحيث يمكن إهمال تأثير التقييد الناجم عن نافذة الاستقبال – بمعنى أن كمية البيانات التي أرسلت ولم يصل إشعار باستلامها بعد عند المُرسِل تُحدّها فقط قيمة المتغير CongWin.

يُحِد القيد أعلاه من كمية البيانات التي يمكن إرسالها بدون انتظار وصول إشعار باستلامها عند المُرسِل، وبالتالي فإنه يُحد بشكلٍ غير مباشر من معدل إرسال البيانات من المُرسِل. لتوضيح ذلك خذ في الاعتبار توصيلة يكون فيها فقد وتأخير الرزم ضئيلين بحيث يمكن إهمالهما. وعليه فإنه تقريباً في بداية كل *RTT* يسمح هذا القيد للمُرسِل بإرسال CongWin بايت من البيانات إلى التوصيلة، وفي نهاية فترة *RTT* يتسلم المُرسِل إشعارات بوصول البيانات. وهكذا فإن معدل إرسال البيانات يساوي تقريباً Congwin/*RTT* بايت/ثانية. باختيار قيمة CongWin يمكن للمُرسِل ضبط المعدل الذي يرسل به البيانات على وصلته.

لنرَ الآن كيف يُدرك مُرسِل TCP أن هناك ازدحاماً على المسار بينه وبين الوجهة. دعنا نعرّف "حدث الفقد" في مُرسِل TCP بحدوث انقضاء فترة موقّت أو تسلم ثلاثة إشعارات استلام مكرّرة من المُستقبل (تذكّر من مناقشتنا في الجزء 3-4-5 لحدث انقضاء فترة الموقّت في الشكل 3-33 والتعديل اللاحق لتضمين الإعادة

السريعة للإرسال عند تلقي ثلاثة إشعارات استلام مكرّرة). عند حدوث ازدحام شديد، يفيض واحد (أو أكثر) من المخازن المؤقتة على الموجّهات على المسار، مما يؤدي إلى سقوط وحدة بيانات (تتضمن قطعة TCP)، وبالتالي ينشأ عن سقوط وحدة البيانات تلك "حدث فقد" لدى المُرسِل – إمّا على شكل انقضاء فترة موقّت أو تلقي ثلاثة إشعارات استلام مكرّرة – والذي يعتبره المُرسِل إشارة لوجود ازدحام على المسار من المُرسِل إلى المُستقبِل.

بعد أن تناولنا كيفية اكتشاف وجود الازدحام، دعنا الآن نرى الحالة الأكثر تفاؤلاً عندما تكون الشبكة لا تعاني من الازدحام، أي عندما لا يكون هناك أحداث فقد. في هذه الحالة سيتلقى مُرسِل TCP إشعارات الاستلام التي ينتظرها لكل القطع التي تم إرسالها. وكما سنرى يأخذ مُرسِل TCP وصول تلك الإشعارات كدليل على أن الأمورعلى ما يرام – أي أن كل القطع التي يتم إرسالها عبر الشبكة تُسلّم بنجاح إلى الوجهة، ومن ثم يزيد قيمة نافذة الازدحام عنده (وبالتالي معدل الإرسال الذي يستخدمه). لاحظ أنه إذا وصلت إشعارات الاستلام بمعدل بطيء (مثلاً بسبب تأخير كبير عبر مسار الشبكة من طرف إلى طرف أو إذا كان ذلك المسار يتضمن وصلة ذات حيّز تردد صغير (أي سعة إرسال منخفضة)، فإن الزيادة في نافذة الازدحام ستكون بمعدل بطيء نسبياً. وبالعكس إذا وصلت إشعارات الاستلام بمعدّل سريع، فسيتم زيادة نافذة الازدحام بسرعة أكبر. نظراً لأن بروتوكول TCP يستخدم إشعارات الاستلام لتحقيق زيادة في نافذة الازدحام (كساعة توقيت)، يُقال عن بروتوكول TCP إنه ذاتي التوقيت.

بوسعنا الآن بحث تفاصيل الخوارزمية التي يستخدمها مُرسِل TCP لضبط معدل إرساله كدالة في الازدحام المحسوس على الشبكة. تلك هي خوارزمية TCP الشهيرة للتحكم في الازدحام. تتألف الخوارزمية من ثلاثة مكونات رئيسة: (1) زيادة خطية ونقصان أسّي، (2) البداية البطيئة، و(3) ردّ الفعل لأحداث انقضاء فترة الموقّت.

الزيادة الخطية والنقصان الأسّي

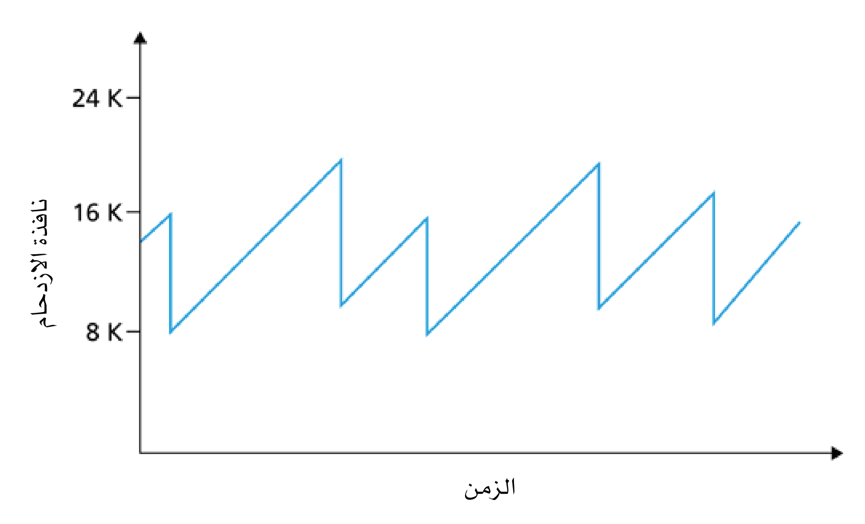
تكمن الفكرة الأساسية وراء التحكم في الازدحام ببروتوكول TCP في تخفيض المُرسِل معدل إرساله (بتصغير نافذة الازدحام CongWin لديه) عند حدوث فقد في الرزم. نظراً لأن توصيلات TCP الأخرى التي تعبر نفس الموجّهات المزدحمة يُحتمل أن تعاني هي الأخرى من "أحداث فقد"، فمن المحتمل أن تقوم هي الأخرى بتخفيض معدلات إرسالها بخفض قيم متغيرات CongWin لديها. وعليه تكون المحصلة النهائية قيام المصادر التي تستخدم المسارات المزدحمة بتخفيض المعدلات التي تستخدمها لإرسال البيانات عبر الشبكة، والذي يُتوقع أن يؤدي بدوره إلى تخفيف حدة الازدحام في المسارات المزدحمة. ولكن ما مقدار التقليل اللازم في نافذة الازدحام على المُرسِل على إثر حدث فقد؟ يستخدم بروتوكول TCP ما يعرف بأسلوب "النقصان الأسّي"، حيث يقلل قيمة CongWin الحالية إلى النصف بعد كل حدث فقد. وبالتالي فإذا كانت قيمة CongWin الحالية على مُرسِل TCP تساوي 20 كيلوبايت وتم اكتشاف حدث فقد، يتم تقليل قيمة CongWin إلى 10 كيلوبايت. إذا وقع حدث فقد آخر، تخفّض CongWin أكثر إلى 5 كيلوبايت. وهكذا يستمر تخفيض قيمة CongWin بحيث لا يُسمح لها بأن تقل عن الحجم الأقصى للقطعة MSS (هذا وصف كلي للصورة الكبيرة لكيفية تغيير نافذة الازدحام بعد حدث فقد. في الواقع فإن الأمر أكثر تعقيداً من ذلك بعض الشيء، كما سنرى قريباً).

بعد أن عرفنا كيف يُخفّض مُرسِل TCP معدل إرساله إزاء اكتشاف ازدحام في الشبكة، من الطبيعي أن نتناول في الخطوة التالية كيف يقوم مُرسِل TCP بزيادة معدل إرساله عندما يدرك أن الشبكة غير مزدحمة، أي عندما يتلقى المُرسِل إشعارات الاستلام المتعلقة بكل القطع التي أُرسلت. إن السبب الجوهري لزيادة معدل الإرسال هو أنه عند عدم وجود ازدحام محسوس يكون هناك احتمال توفر حيّز تردد غير مستعمل يمكن استغلاله لصالح توصيلة TCP. في مثل هذه الظروف يزيد مُرسِل TCP من نافذة الازدحام لديه ببطء، متقصّياً بحذر الحيّز الترددي الإضافي المتوفر على المسار من طرف إلى طرف. يقوم مُرسِل TCP بذلك بزيادة قيمة

نافذة الازدحام رويداً رويداً في كل مرة يتلقى فيها إشعار استلام بهدف زيادتها بـ MSS واحدة في كل *RTT* [RFC 2581]، ويمكن تحقيق ذلك بعدة طرق.

من الطرق المستخدمة بكثرة أن يقوم المُرسِل بزيادة CongWin بمقدار (MSS×MSS/CongWin) بايت في كل مرة يصله فيها إشعار استلام جديد. على سبيل المثال إذا كانت قيمة *MSS* تساوي 1,460 بايتاً وقيمة CongWin تساوي 14,600 بايت، يتم إرسال 10 قطع خلال وقت رحلة الذهاب والعودة *RTT*، ويؤدي وصول كل إشعار استلام (بافتراض استخدام إشعار استلام لكل قطعة) إلى زيادة نافذة الازدحام بمقدار *MSS*/10، وعليه فبعد وصول إشعارات الاستلام للقطع العشر التي أُرسلت خلال *RTT*، تكون نافذة الازدحام قد زادت بمقدار *MSS*، وهو المطلوب.

الخلاصة: إنّ مُرسِل TCP يزيد من معدل إرساله بطريقة الإضافة عندما يدرك أن المسار من طرف إلى طرف خالٍ من الازدحام، ويُنقص المعدل بطريقة أسّية عندما يكتشف (عن طريق حدث فقد للرزم) أن المسار مزدحم. لهذا السبب غالباً ما يعرف أسلوب التحكم في الازدحام في بروتوكول TCP بأنه خوارزمية من نوع الزيادة الخطية والنقصان الأسّي ((Additive Increase Multiplicative Decrease (AIMD)). تُعرف مرحلة الزيادة الخطية في عملية التحكم في الازدحام ببروتوكول TCP بمرحلة تجنب الازدحام. تمر قيمة CongWin بدورة متكرّرة تبدأ فيها بالزيادة الخطية ثم تهبط فجأة إلى نصف قيمتها الحالية (عند وقوع حدث فقد)، ومن ثم تأخذ التغيرات في تلك القيمة شكل سن المنشار في توصيلات TCP طويلة الأمد، كما هو مبين في الشكل 3-51.

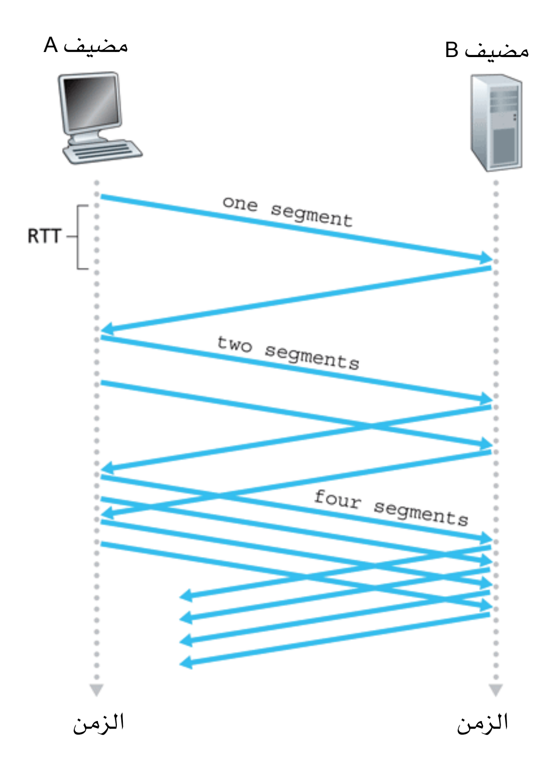


الشكل 3-51 التحكم في الازدحام بزيادة خطية ونقص أسّي.

البداية البطيئة (Slow Start)

عندما تبدأ توصيلة TCP في العمل تأخذ CongWin قيمة مبدئية تكون عادةً مساوية لـ MSS واحدة [RFC 3390]، ومن ثم يكون معدل الإرسال الأوّلي MSS/RTT تقريباً. كمثال إذا كانت MSS = 500 بايت وRTT = 200 ميللي ثانية، فإن معدل الإرسال الناتج يكون 20 كيلوبت/ثانية. نظراً لأن حيّز التردد المتوفر للوصلة قد يكون أكثر بكثير من *MSS/RTT* يكون من المؤسف زيادة معدل الإرسال فقط بشكلٍ خطي والانتظار لوقت طويل أكثر مما ينبغي حتى يرتفع معدل الإرسال إلى مستوى معقول. وعليه فبدلاً من زيادة المعدل بشكلٍ خطي أثناء تلك المرحلة الأولية، يقوم مُرسِل TCP بزيادة المعدل تصاعدياً (بشكلٍ أسّي) بمضاعفة قيمة CongWin كل *RTT*. يواصل مُرسِل TCP زيادة معدل الإرسال بطريقة أسّية حتى يقع حدث فقد، عندئذٍ تُنصّف قيمة CongWin وبعد ذلك تنمو بشكلٍ خطي، كما سبق وصفه من قبل. وهكذا فأثناء تلك المرحلة الأولية التي تدعى البداية البطيئة (Slow Start (SS))، يبدأ مُرسِل TCP الإرسال بمعدل بطيء (ومن هنا كانت التسمية البداية البطيئة) ولكنه يزيد معدل الإرسال بطريقة أسّية، يحدث ذلك بزيادة المُرسِل قيمة CongWin في كل مرة بـ MSS واحدة عن

كل قطعة تم إرسالها وتلقّى المُرسِل إشعاراً باستلامها. كما هو مبين في الشكل 3-52 يرسل TCP القطعة الأولى إلى الشبكة وينتظر إشعار استلام. إذا وصل إشعار باستلام تلك القطعة قبل وقوع حدث فقد، يزيد مُرسِل TCP نافذة الازدحام بـ MSS واحدة وبالتالي يبعث بقطعتين بأقصى حجم ممكن. إذا وصل إشعار استلام لهذين القطعتين قبل وقوع حدث فقد، يزيد المُرسِل نافذة الازدحام بـ MSS لكل واحدة من هاتين القطعتين اللتين تم الإشعار باستلامهما، ومن ثم تصبح نافذة الازدحام 4×MSS ويقوم المُرسِل ببث أربع قطع بأقصى حجم ممكن. يستمر هذا الإجراء طالما استمرت إشعارات الاستلام في الوصول، إلى أن يقع حدث فقد في نهاية الأمر. وهكذا تتضاعف عملياً قيمة CongWin كل *RTT* أثناء مرحلة البداية البطيئة.



الشكل 3-52 البداية البطيئة لبروتوكول TCP

ردّ الفعل لأحداث انقضاء فترة الموقّت

تتلخص الصورة التي رسمناها حتى الآن لنافذة الازدحام ببروتوكول TCP في النمو الأسي بدءاً من MSS واحدة (أثناء مرحلة البداية البطيئة) إلى حين وقوع حدث فقد، حيث يبدأ عندئذٍ نمط سن المنشار. وعلى الرغم من أن هذه الصورة قريبة من وصف الواقع بدقة، إلا أننا سنكون مقصرين إذا لم نذكر أن أسلوب التحكم في الازدحام ببروتوكول TCP يستجيب في الواقع لحدث الفقد الذي يُكتشف بانقضاء فترة موقّت بشكلٍ مختلف عن حدث الفقد الذي يُكتشف بوصول ثلاثة إشعارات استلام مكرّرة. فبعد وصول ثلاثة إشعارات استلام مكرّرة، يتصرّف بروتوكول TCP بالطريقة التي وصفناها آنفاً، حيث تتصّف نافذة الازدحام وبعد ذلك تزيد قيمتها بشكلٍ خطي. أما بعد وقوع حدث انقضاء فترة موقّت، فإن مُرسِل TCP يدخل مرحلة البداية البطيئة، أي يضبط قيمة نافذة الازدحام عند MSS واحدة وبعد ذلك تنمو قيمة النافذة بطريقة أسّية. تواصل قيمة النافذة نموها التصاعدي حتى تصل CongWin إلى نصف قيمتها قبل وقوع حدث انقضاء فترة الموقّت مباشرة. عند تلك النقطة، تبدأ CongWin في النمو بشكلٍ خطي، تماماً كما هو الحال بعد تلقي ثلاثة إشعارات استلام مكرّرة.

يدير بروتوكول TCP هذه الديناميكية المعقدة بالاحتفاظ بمتغير يسمى العتبة (threshold) يحدد قيمة نافذة الازدحام التي تنتهي عندها مرحلة البداية البطيئة وتبدأ مرحلة تجنب الازدحام. في البداية يأخذ متغير العتبة قيمة كبيرة (65 كيلوبايت عملياً [Stevens 1994]) بحيث لا يكون له تأثير أولي. وعند وقوع حدث فقد، توضع قيمة العتبة مساويةً لنصف القيمة الحالية لـ CongWin. على سبيل المثال، إذا كانت قيمة نافذة الازدحام 20 كيلوبايت مباشرةً قبل وقوع حدث فقد، فإن قيمة العتبة تصبح 10 كيلوبايت، وتبقى هذه القيمة لحين وقوع حدث الفقد التالي.

بعد أن وصفنا متغير العتبة، بوسعنا الآن أن نصف بالضبط كيف يتصرّف المتغير CongWin بعد حدث انقضاء فترة موقّت. كما أشرنا من قبل يدخل مُرسِل

TCP مرحلة البداية البطيئة بعد حدث انقضاء فترة موقّت. أثناء مرحلة البداية البطيئة يتم زيادة قيمة CongWin بطريقة أسّية بسرعة حتى تصل CongWin إلى العتبة، عند ذلك يدخل TCP مرحلة تجنب الازدحام، وخلالها تزداد قيمة CongWin بشكلٍ خطي كما وصفنا في وقت سابق.

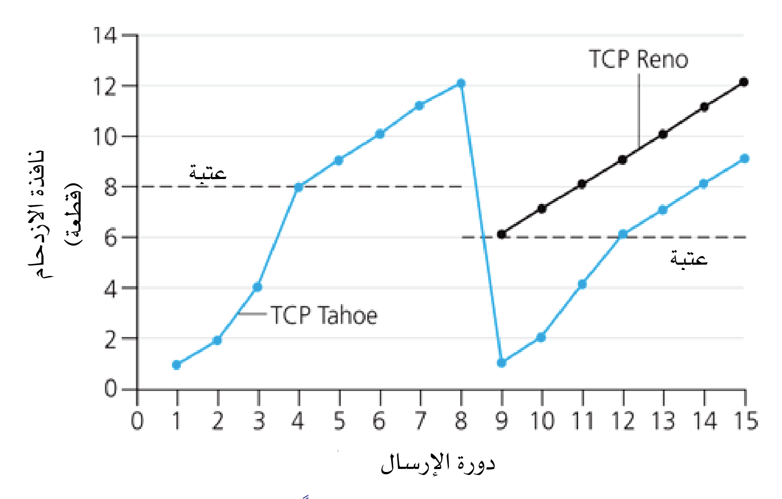
يلخص الجدول 3-3 مناقشتنا لخوارزمية التحكم في الازدحام ببروتوكول TCP. عند هذه النقطة من الطبيعي أن نتساءل لماذا تتصرّف الخوارزمية بشكلٍ مختلف بعد حدث انقضاء فترة موقّت عنه بعد تلقي ثلاثة إشعارات استلام مكرّرة. بالتحديد، ما الذي يجعل مُرسِل TCP يتصرّف بتحفظ أكثر إثر وقوع حدث انقضاء فترة موقّت، فيضع قيمة نافذة الازدحام عند MSS واحدة، بينما عند تلقّي ثلاثة إشعارات استلام مكرّرة يكتفي فقط بتقليل نافذة الازدحام إلى النصف؟ من الغريب أن إصداراً مبكراً من بروتوكول TCP، يعرف ببروتوكول TCP Tahoe، يُحِد من نافذة الازدحام عند MSS واحدة بشكلٍ غير مشروط ويدخل مرحلة البداية البطيئة بعد أيٍّ من نوعي حدث الفقد. أما الإصدار الأحدث TCP Reno فيلغي مرحلة البداية البطيئة بعد تلقي ثلاثة إشعارات استلام مكرّرة. إنّ الفلسفة وراء إلغاء البداية البطيئة في تلك الحالة الأخيرة أنه بالرغم من أن رزمة قد فقدت، إلا أن وصول ثلاثة إشعارات استلام مكرّرة يدل على أن المُستقبِل قد استلم بعض القطع (بالتحديد ثلاث قطع إضافية بعد القطعة المفقودة). وهكذا فبخلاف حالة انقضاء فترة الموقّت، تبدو الشبكة قادرة على توصيل بعض القطع على الأقل، حتى إذا كانت هناك قطع أخرى تُفقد بسبب الازدحام. هذا الإلغاء لمرحلة البداية البطيئة بعد ثلاثة إشعارات استلام مكرّرة يعرف بالتعافي السريع (fast recovery) من أثر الازدحام.

الحالة	الحدث	إجراء التحكم في الازدحام في مُرسِل TCP	ملاحظات
بداية بطيئة (SS)	وصول إشعار استلام بيانات لم يتم الإشعار باستلامها من قبل	CongWin = CongWin + MSS, If (CongWin > Threshold) set state to "Congestion Avoidance" انتقل إلى حالة "تجنب الازدحام"	ينتج عنه مضاعفة CongWin كل *RTT*
تجنب الازدحام (CA)	وصول إشعار استلام بيانات لم يتم الإشعار باستلامها من قبل	CongWin=CongWin+ MSS × (MSS/CongWin)	زيادة خطيّة، ينتج عنها زيادة CongWin بـ MSS واحدة كل *RTT*.
SS أو CA	اكتشاف حدث فقد بإشعارات استلام ثلاثية مكرّرة	Threshold=CongWin/2 CongWin=Threshold انتقل إلى حالة "تجنب الازدحام"	تعافٍ سريع، تفعيل التنقيص الضربي. لن تقل CongWin عن MSS واحدة
SS أو CA	انقضاء فترة الموقّت	Threshold=CongWin/2 CongWin =1 MSS انتقل إلى حالة "البداية البطيئة"	دخول مرحلة البداية البطيئة
SS أو CA	إشعارات استلام مكرّرة	قم بزيادة عدد إشعارات الاستلام المكرّرة للقطعة الجاري الإشعار باستلامها	تبقى كلٌّ من CongWin وThreshold بدون تغيير.

الجدول 3-3 التحكم في الازدحام لدى مُرسِل TCP، على افتراض أن قيمة CongWin الأولية تساوي MSS، والقيمة الأولية للعتبة (threshold) كبيرة (مثلاً 65 كيلوبايت [Stevens 1994])، وأن مُرسِل TCP يبدأ في حالة البداية البطيئة. الحالة المبينة بالجدول هي حالة مُرسِل TCP قبل حصول الحدث مباشرةً. راجع [RFC 2581] لمزيد من التفاصيل.

يبين الشكل 3-53 التغير في قيمة نافذة الازدحام لكلٍ من إصداري Reno وTahoe من بروتوكول TCP، في هذا الشكل كانت القيمة الأولية للعتبة 8×MSS لدورات الإرسال الثماني الأولى يتّخذ كلٌّ من Reno وTahoe إجراءات مماثلة. تزداد قيمة نافذة الازدحام بطريقة أسّية بسرعة أثناء مرحلة البداية البطيئة حتى تلتقي

بالعتبة في الدورة الرابعة للإرسال. بعد ذلك تزداد قيمة نافذة الازدحام بشكلٍ خطي إلى أن يحدث إشعار استلام ثلاثي مكرّر مباشرةً بعد دورة الإرسال رقم 8. لاحظ أن قيمة نافذة الازدحام كانت 12×MSS عند وقوع حدث الفقد هذا. عندئذ تضبط قيمة العتبة عند نصف CongWin أي تصبح 6×MSS. تبعاً لإصدار TCP Reno، تأخذ نافذة الازدحام القيمة 6×MSS وبعد ذلك تنمو بشكلٍ خطي. تبعاً لإصدار TCP Tahoe تأخذ نافذة الازدحام القيمة MSS واحدة وبعد ذلك تنمو بشكلٍ أسّي حتى تصل إلى العتبة. هذه الخوارزمية للتحكم في الازدحام من تأليف Jacobson [Jacobson 1988]، يوجد وصف لعدد من التعديلات على خوارزمية Jacobson الأصلية في [Stevens 1994] وفي [RFC 2561].



الشكل 3-53 نموّ نافذة الازدحام لبروتوكول TCP تبعاً للخوارزميتين Tahoe و Reno.

كما ذكرنا آنفاً تستخدم أكثر تطبيقات بروتوكول TCP الحالية خوارزمية Reno. تم اقتراح العديد من نوعيات خوارزمية Reno [RFC 3782; RFC 2018]. تسعى خوارزمية Vegas المقترحة [Brakmo 1995; Ahn 1995] لتفادي الازدحام مع الحفاظ على طاقة إنتاجية عالية. الفكرة الأساسية وراء خوارزمية Vegas هي: (1) اكتشاف الازدحام في الموجّهات بين المصدر والوجهة النهائية قبل حدوث فقد

للرزم، و(2) تخفيض معدل الإرسال بشكلٍ خطي عند اكتشاف هذا الفقد الوشيك للرزم. يتم توقع الفقد الوشيك للرزم بملاحظة وقت رحلة الذهاب والإياب *RTT*، فكلما كان *RTT* للرزم أطول كان الازدحام في الموجّهات أكبر.

وصف ماكروسكوبي (تقريبي) للطاقة الإنتاجية لبروتوكول TCP

بالنظر إلى نمط سن المنشار الذي يميز سلوك بروتوكول TCP في التحكم في الازدحام، من الطبيعي أن نفكر في حساب الطاقة الإنتاجية المتوسطة (أي معدل الإرسال المتوسط) لتوصيلة TCP طويلة الأمد. في هذا التحليل سنهمل مراحل البداية البطيئة التي تحدث بعد أحداث انقضاء فترة الموقّت. (عادةً ما تكون هذه المراحل قصيرة جداً، حيث تنمو قيمة نافذة الازدحام لدى المُرسِل بطريقة أسّية ويتم تجاوز تلك المرحلة بسرعة). أثناء فترة رحلة ذهاب وإياب معينة، يعتمد المعدل الذي يرسل به بروتوكول TCP البيانات على نافذة الازدحام وقيمة *RTT* الحالية. عندما تكون نافذة الازدحام *w* بايتات ووقت رحلة الذهاب والإياب الحالي *RTT* ثانية، يكون معدل الإرسال تقريباً *w*/*RTT*. يقوم TCP بعد ذلك باستكشاف وجود حيز تردد إضافي، وذلك بزيادة *w* بـ MSS واحدة كل فترة *RTT* إلى أن يقع حدث فقد. لنرمز لقيمة النافذة عند وقوع حدث الفقد بالرمز *W*. على افتراض أن كلاً من *RTT* و*W* ثابتان تقريباً طوال مدة التوصيلة، فإن معدل إرسال TCP يتراوح من $W/(2\times RTT)$ إلى W/RTT.

تؤدي هذه الفرضيات إلى نموذج ماكروسكوبي (تقريبي) مبسط للغاية لسلوك بروتوكول TCP في الحالة الاستقرار (steady state). تُسقط الشبكة رزمة للتوصيلة عندما يزداد معدل الإرسال إلى *W*/*RTT*، وعندها يقلل المُرسِل معدل إرساله إلى النصف ثم يعاود زيادته بمقدار *MSS*/*RTT* كل فترة *RTT* إلى أن يصل من جديد إلى *W*/*RTT*. تكرّر هذه العملية نفسها مراراً وتكراراً. نظراً لأن طاقة TCP الإنتاجية (أي معدل الإرسال) تزيد بشكلٍ خطي بين هاتين القيمتين الطرفيتين، فإننا نحصل على:

$$\text{معدل الإرسال المتوسط للتوصيلة} = \frac{0.75\,W}{RTT}$$

باستعمال هذا النموذج المثالي جداً لديناميكية بروتوكول TCP في حالة الاستقرار (steady state)، يمكننا أن نشتق تعبيراً شائقاً يربط معدل فقد الرزم على التوصيلة بحيز التردد المتاح لها [Mahdavi 1997]، وسيتم تناول ذلك من خلال التمارين. هناك نموذج أكثر تطوّراً تم التوصل إليه بشكلٍ تجريبي ليوافق قياسات البيانات [Padhye 2000].

مُستقبِل بروتوكول TCP

من المهم إدراك أن أساليب التحكم في الازدحام في بروتوكول TCP قد تطورت على مر السنين، وهي تواصل تطورها حالياً. يمكن الاطلاع على ملخص للتحكم في الازدحام في TCP ابتداءً من أواخر التسعينيات في [RFC 2581]. لاستعراض التطوّرات الأخرى هذا المجال راجع [Floyd 2001]. واضحٌ أن ما كان مناسباً للإنترنت عندما كانت توصيلات TCP تنقل في الغالب حركة مرور بيانات SMTP وFTP وTelnet لن تكون بالضرورة مناسبة لإنترنت اليوم التي يغلب عليها حركة مرور بيانات HTTP أو لإنترنت المُستقبِل بخدمات لم نحلم بها بعد.

لتوضيح الحاجة المستمرة لتطوير بروتوكول TCP، دعنا نأخذ بعين الاعتبار وصلات TCP السريعة المطلوبة لتطبيقات الحوسبة الشبكية (grid computing) [Foster 2002]. خذ على سبيل المثال توصيلة TCP بقطع بيانات مقاسها 1,500 بايت ووقت رحلة الذهاب والإياب قدره 100 ميللي ثانية. افترض أننا نريد إرسال البيانات عبر هذه التوصيلة بمعدل 10 جيجابت/ثانية. تبعاً لـ [RFC 3649] نلاحظ أنه باستخدام معادلة الطاقة الإنتاجية المذكورة آنفاً، لتحقيق طاقة إنتاجية قدرها 10 جيجابت/ثانية ينبغي استخدام نافذة ازدحام مقاسها 83,333 قطعة في المتوسط. تلك كمية كبيرة من القطع، مما يجعلنا قلقين بعض الشيء بخصوص ما يمكن أن يحدث إذا فقدت إحدى تلك القطع الـ 83,333 أثناء انتقالها. ماذا يحدث في حالة الفقد؟ بمعنى آخر، ما النسبة التي يمكن أن تفقد من القطع المُرسَلة ومع ذلك تسمح لخوارزمية TCP للتحكم في الازدحام والملخّصة في الجدول 3-3 بإنجاز معدل الإرسال 10 جيجابت/ثانية المطلوب؟ في تمارين هذا الفصل سيتم توجيهك عبر

خطوات اشتقاق معادلة لحساب الطاقة الإنتاجية لتوصيلة TCP بدلالة نسبة الفقد L، وقت رحلة الذهاب والإياب RTT، والحجم الأقصى للقطعة MSS:

$$\text{معدل الإرسال المتوسط للتوصيلة} = \frac{1.22\,MSS}{RTT\sqrt{L}}$$

باستعمال هذه المعادلة يمكننا أن نرى أنه لكي نحقق طاقة إنتاجية مقدارها 10 جيجابت/ثانية، يمكن أن تتحمل خوارزمية التحكم في الازدحام احتمال فقد قطعة بحد أقصى 2×10^{-10} (أو مايعادل حدث فقد واحد لكل 5,000,000,000 قطعة ترسل؛ وهي نسبة فقد منخفضة جداً). أدت هذه الملاحظة بعدد من الباحثين للبحث عن إصدارات جديدة من بروتوكول TCP مصممة خصّيصاً لمثل هذه البيئات للنقل السريع للبيانات، يمكنك مراجعة [Jin 2004; RFC 3649; Kelly 2003] لمناقشة تلك الجهود.

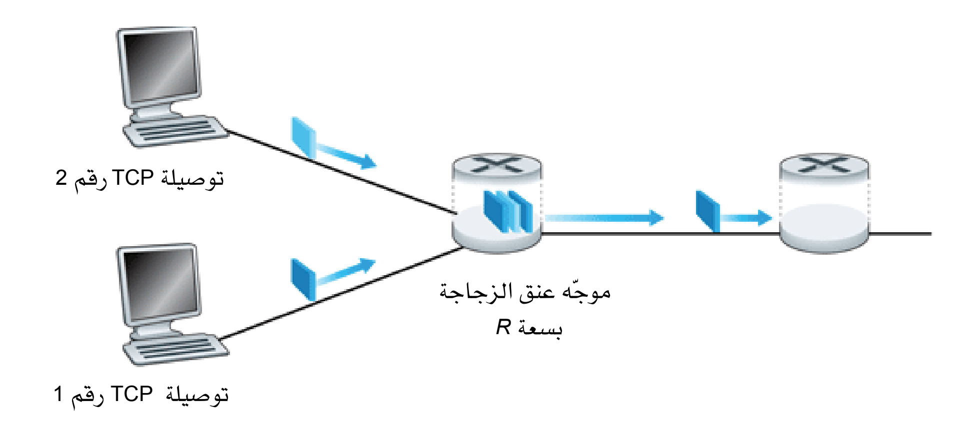
3-7-1 عدالة توزيع سعة الإرسال

خذ في الاعتبار عدد K من توصيلات TCP لكلٍّ منها مسار من طرف إلى طرف، ولكنها تمر جميعاً عبر وصلة عنق الزجاجة بمعدل إرسال قدره R بت/ثانية (نعني بوصلة عنق الزجاجة أن كل الوصلات الأخرى على مسار كل توصيلة ليست مزدحمة وتتوافر لديها سعة إرسال كافية مقارنةً بسعة الإرسال لوصلة عنق الزجاجة). افترض أن كل توصيلة تنقل ملفاً كبيراً وأنه لا توجد حركة بيانات UDP تمر عبر وصلة عنق الزجاجة. يقال: إن آلية التحكم في الازدحام عادلة إذا كان معدل الإرسال المتوسط لكل توصيلة يساوي تقريباً R/K، أي أن كل توصيلة تحصل على نصيب متساوٍ من حيّز التردد للوصلة.

السؤال هو هل خوارزمية AIMD ببروتوكول TCP عادلة، علماً بأن توصيلات TCP المختلفة يمكن أن تبدأ في أوقات مختلفة ومن ثم يمكن أن تكون لها نوافذ ازدحام بقيم مختلفة في نقطة معينة من الزمن؟ يتضمن [Chiu 1989] تفسيراً بديهياً رائعاً يوضح كيف يتقارب أسلوب التحكم في الازدحام ببروتوكول TCP بمرور

الوقت ليوفر حصة متساوية من حيّز التردد لوصلة عنق الزجاجة لكلٍ من توصيلات TCP المتنافسة.

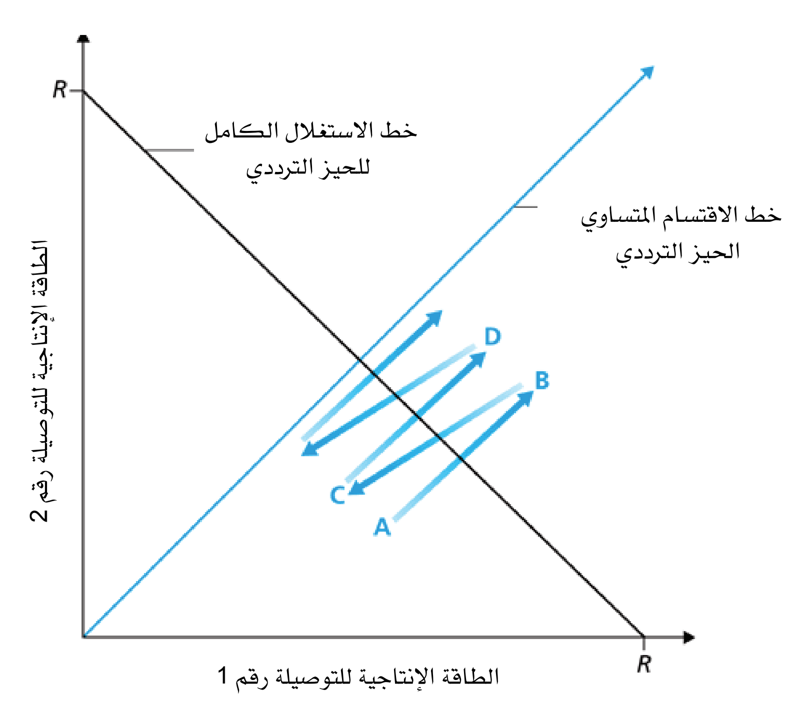
لنأخذ في الاعتبار الحالة البسيطة لتوصيلتي TCP تشتركان في وصلة واحدة لها معدل إرسال مقداره R كما هو مبيّن في الشكل 3-54. افترض أن كلتا التوصيلتين لهما نفس القيم للمتغيرات MSS وRTT (بحيث إنه إذا كان لهما نفس حجم نافذة الازدحام، فسيكون لهما نفس الطاقة الإنتاجية)، وأن لديهما كمية كبيرة من البيانات المطلوب إرسالها، وأنه لا توجد توصيلات TCP أخرى أو وحدات بيانات UDP تعبر تلك الوصلة المشتركة. افترض أيضاً أننا سنهمل مرحلة البداية البطيئة لبروتوكول TCP وأن كل توصيلات TCP تعمل في نمط تجنب الازدحام (AIMD) في جميع الأوقات.



الشكل 3-54 توصيلتا TCP مشتركتان في وصلة عنق زجاجة واحدة.

يوضح المخطط البياني بالشكل 3-55 الطاقة الإنتاجية التي تحصّلها كلٌّ من توصيلتى TCP. لتحقيق مشاركة عادلة بين التوصيلتين في الحيّز الترددي للوصلة المشتركة، ينبغي أن تقع الطاقة الإنتاجية المتحققة على خط ينطلق من نقطة الأصل بميل 45 درجة. مثالياً يجب أن يساوي مجموع الطاقتين الإنتاجيتين للتوصيلتين سعة إرسال الوصلة المشتركة R. (بالتأكيد يُعتبر حصول كل توصيلة على حصة

متساوية تساوي الصفر من سعة الإرسال للوصلة يُعتبر أمراً غير مرغوب فيه!). وعليه يجب أن يكون الهدف هو أن تقع الطاقة الإنتاجية المنجزة في مكان ما بالقرب من تقاطع خط المشاركة المتساوية في الحيّز الترددي (بميل 45 درجة) وخط الاستغلال الكامل للحيّز الترددي في الشكل 3-55.



الشكل 3-55 الطاقة الإنتاجية المتحققة لتوصيلتي TCP رقم 1 و2.

افترض أن مقاسات نوافذ TCP في نقطة معينة من الزمن كانت بحيث تمثّل الطاقة الإنتاجية لكلٍ من التوصيلتين 1 و2 بالنقطة A في الشكل 3-55. نظراً لأن سعة إرسال الوصلة المستخدمة بالتوصيلتين معاً أقل من R عند هذه النقطة، فلن يحدث فقد للرزم وستقوم كلٌّ من التوصيلتين بزيادة مقاس نافذتها بمعدل MSS واحدة لكل فترة RTT حسب خوارزمية TCP لتجنب الازدحام. وهكذا تسير الطاقة الإنتاجية المشتركة للتوصيلتين على خط بميل 45 درجة (زيادة متساوية لكلتا التوصيلتين) بدءاً من النقطة A. في النهاية ستتجاوز سعة إرسال الوصلة المستخدمة

بالتوصيلتين معاً القيمة R، ويبدأ وقوع أحداث فقد للرزم. افترض أن التوصيلتين 1 و2 تعانيان من فقد رزم عندما تكون الطاقة الإنتاجية لهما ممثلة بالنقطة B. عندئذ ستقرر التوصيلتان 1 و2 إنقاص نوافذهما بمقدار النصف، ومن ثم تصبح الطاقات الإنتاجية الناتجة ممثلة بالنقطة C في منتصف المسافة على الخط الواصل من B إلى نقطة الأصل. نظراً لأن سعة إرسال الوصلة المستخدمة بالتوصيلتين معاً أقل من R عند النقطة C، تزيد التوصيلتان من طاقتيهما الإنتاجية مرة أخرى على طول خط بميل 45 درجة يبدأ من C. في النهاية سيحدث فقد للرزم من جديد، على سبيل المثال عند النقطة D، وعندئذٍ تُنقص التوصيلتان نوافذهما ثانيةً بمقدار النصف. وهكذا عليك إقناع نفسك بأن معدل الإرسال المتحقق للتوصيلتين سيتأرجح في النهاية على طول خط معدل الإرسال المتساوي. كما ينبغي أن تقنع نفسك أيضاً بأن التوصيلتين ستصلان تقاربياً لهذا السلوك بغض النظر عن موقعهما في البداية في الفضاء ثنائي الأبعاد! رغم أن هذا السيناريو يعتمد على عددٍ من الفرضيات المثالية، إلا أنه يعطينا إحساساً بديهياً للسبب وراء توفير بروتوكول TCP مشاركة متساوية في حيّز التردد للوصلة المشتركة بين التوصيلات التي تستخدمها.

في السيناريو المثالي الذي نحن بصدده افترضنا أن توصيلات TCP فقط هي التي تعبر وصلة عنق الزجاجة، وأن تلك التوصيلات لها نفس قيمة RTT، وأن هناك توصيلة TCP واحدة مرتبطة بكل زوج من مضيفات المصدر والوجهة النهائية. في الواقع العملي لا تتحقق تلك الشروط عادةً، ومن ثم يمكن أن تحصل تطبيقات الزبون/الخادم على أنصبة غير متساوية أبداً من الحيّز الترددي للوصلة المشتركة. بشكلٍ خاص تم إثبات أنه عند اشتراك عدة توصيلات في وصلة عنق الزجاجة فإن الجلسات التي لها قيم RTT أصغر تتمكن من الاستحواذ على الحيّز الترددي على تلك الوصلة بسرعة أكبر عند توفره (أي تقوم بفتح نوافذ الازدحام الخاصة بها بشكلٍ أسرع) ومن ثم تتمتع بطاقة إنتاجية أعلى من تلك التوصيلات التي لها قيم RTT أكبر [Lakshman 1997].

بروتوكول UDP وعدالة توزيع سعة الإرسال

رأينا الآن كيف أن التحكم ﮐ الازدحام ﮐ بروتوكول TCP ينظم معدل الإرسال من التطبيقات عن طريق آلية نافذة الازدحام. لا يستخدم العديد من تطبيقات الوسائط المتعددة، كهاتف الإنترنت والمؤتمرات عبر الفيديو بروتوكول TCP ﮐ أغلب الأحيان لهذا السبب بعينه – فهي لا تقبل بخنق معدل الإرسال لها حتى إذا كانت الشبكة مزدحمة جداً. بدلاً من ذلك تفضّل تلك التطبيقات التشغيل على بروتوكول UDP، والذي لا يتضمن سيطرة مبيّتة على الازدحام. عند تشغيلها على UDP يمكن للتطبيقات ضخ بيانات الصوت والفيديو إلى الشبكة بمعدل إرسال ثابت وتفقد الرزم من حين لآخر، وهي تفضل ذلك على تقليل معدلات إرسالها إلى مستويات "عادلة" ﮐ أوقات الازدحام مع عدم فقد أي رزم. من منظور بروتوكول TCP، لا تعتبر تطبيقات الوسائط المتعددة التي يتم تشغيلها على UDP منصفة – فهي لا تتعاون مع التوصيلات الأخرى ولا تتحكم ﮐ معدلات إرسالها بشكلٍ ملائم. نظراً لأن التحكم ﮐ الازدحام ﮐ بروتوكول TCP يقلل من معدل الإرسال لمواجهة الازدحام المتزايد (فقد الرزم)، بينما لا تحتاج مصادر UDP لذلك، فمن المحتمل أن تزاحم مصادر UDP حركة مرور TCP وتضيّق المجال عليها. وعليه فمن مجالات البحث المهمة اليوم تطوير آليات للتحكم ﮐ الازدحام بالإنترنت تمنع حركة مرور UDP من جعل طاقة الإنترنت الإنتاجية تتدهور لتصل إلى توقّف كامل [Floyd 1999; Floyd 2000; Kohler 2006].

عدالة توزيع سعة الإرسال وتوصيلات TCP المتوازية

لكن حتى لو أمكننا إجبار حركة مرور UDP على التصرّف بإنصاف، فإن مشكلة عدالة التوزيع لن تكون قد حُلّت تماماً. يكمن السبب ﮐ ذلك ﮐ أنه ليس هناك ما يمنع التطبيقات المبنية على بروتوكول TCP من استخدام عدة توصيلات على التوازي. على سبيل المثال غالباً ما تستخدم متصفحات الويب عدة توصيلات TCP متوازية لنقل بيانات عدة كائنات ضمن صفحة الويب ﮐ نفس الوقت (العدد الدقيق للتوصيلات المتعدّدة هو متغير قابل للضبط ﮐ أكثر المتصفّحات). عندما

يستخدم تطبيقٌ ما عدة توصيلات متوازية فإنه يستحوذ على جزء أكبر من حيّز التردد لوصلة مزدحمة. كمثال خذ في الاعتبار وصلة بسعة إرسال R تدعم تسعة تطبيقات زبون/خادم مستمرة، بحيث يستخدم كل تطبيق توصيلة TCP واحدة. إذا ظهر تطبيق جديد يستخدم توصيلة TCP واحدة كذلك، فإن كل تطبيق سيستخدم تقريباً نفس قيمة معدل الإرسال والتي مقدارها $R/10$. أما إذا كان ذلك التطبيق الجديد يستخدم بدلاً من ذلك 11 توصيلة TCP متوازية، فإن التطبيق الجديد سيستحوذ على نصيب غير عادل يزيد على $R/2$. نظراً لأن حركة مرور الويب واسعة الانتشار جداً على الإنترنت، فإن توصيلات TCP المتعدّدة على التوازي تعد أمراً شائعاً.

3-8 الخلاصة

بدأنا هذا الفصل بدراسة الخدمات التي يوفرها بروتوكول طبقة النقل لتطبيقات الشبكة. فمن ناحية قد يكون بروتوكول طبقة النقل بسيطاً للغاية بحيث يوفر خدمة بلا رتوش للتطبيقات التي تستخدمه، فيقوم فقط بوظيفة التجميع والتوزيع للعمليات التي تتصل فيما بينها. ومثال ذلك بروتوكول الإنترنت UDP. وعلى الطرف الآخر يمكن أن يوفر بروتوكول طبقة النقل تشكيلة من الضمانات للتطبيقات، كالتوصيل الموثوق للبيانات، وضمانات التأخير، وضمانات الحيّز الترددي. ومع ذلك فغالباً ما يُحِد نموذج الخدمة لبروتوكول طبقة الشبكة التحتي من الخدمات التي يمكن أن يوفرها بروتوكول طبقة النقل. فمثلاً إذا كان بروتوكول طبقة الشبكة لا يستطيع توفير ضمانات الحيّز الترددي أو التأخير لقطع طبقة النقل، فإن بروتوكول طبقة النقل لن يستطيع توفير ضمانات الحيّز الترددي أو التأخير لقطع البيانات التي تُنقل بين العمليات.

عرفنا في الجزء 3-4 أن بروتوكول طبقة النقل يمكن أن يوفر نقلاً موثوقاً للبيانات حتى لو كانت طبقة الشبكة التحتية غير موثوقة. ورأينا أن توفير نقلٍ موثوقٍ للبيانات يتضمن العديد من النقاط الدقيقة، لكن هذه المهمة يمكن أن

تنجز من خلال الجمع بعناية ما بين إشعارات الاستلام، والموقّتات، وإعادة الإرسال، والأرقام التسلسلية.

رغم أننا غطّينا النقل الموثوق للبيانات في هذا الفصل، يجب ألا يغيب عن بالنا أن النقل الموثوق للبيانات يمكن أن توفره بروتوكولات طبقة ربط البيانات أو طبقة الشبكة أو طبقة النقل أو حتى طبقة التطبيقات. يمكن لأيٍ من الطبقات العليا الأربعة في رصة البروتوكولات أن تستخدم إشعارات الاستلام، والموقّتات، وإعادة الإرسال، والأرقام التسلسلية، لتوفر نقلاً موثوقاً للبيانات للطبقة التي تعلوها. في الحقيقة وعلى مر السنين صمّم علماء ومهندسو الحاسب وطوروا بشكلٍ مستقل بروتوكولات للنقل الموثوق للبيانات لطبقة ربط البيانات، وطبقة الشبكة، وطبقة النقل، وطبقة التطبيقات (رغم أن العديد من تلك البروتوكولات اختفت بهدوء من الساحة).

في الجزء 3-5 ألقينا نظرة فاحصة على بروتوكول TCP، بروتوكول الإنترنت التوصيلي للنقل الموثوق للبيانات. عرفنا أن بروتوكول TCP نظام معقّد، يتضمّن إدارة التوصيلات، وضبط التدفق، وتقدير وقت رحلة الذهاب والإياب، بالإضافة إلى النقل الموثوق للبيانات. ومع ذلك يُعتبر بروتوكول TCP في الواقع أكثر تعقيداً من الوصف الذي قدمناه. لقد أغفلنا عمداً مناقشة تشكيلة من الترقيعات والإصلاحات والتحسينات التي شاع استخدامها على نطاق واسع في إصدارات بروتوكول TCP المختلفة. ومع ذلك فإن كل هذا التعقيد يتم حجبه عن التطبيق الذي يتم تشغيله على الشبكة. إذا أراد زبون على مضيف إرسال البيانات بشكلٍ موثوق إلى خادم على مضيف آخر، فإنه يفتح ببساطة مقبس TCP على الخادم ويضخ البيانات عبر ذلك المقبس. إن تطبيق الزبون/الخادم محظوظ لأنه لا يحمل هم أيٍّ من التعقيد الذي ينطوي عليه بروتوكول TCP.

استعرضنا في الجزء 3-6 موضوع التحكم في الازدحام من منظور واسع، وفي الجزء 3-7 شرحنا كيف يطبّق بروتوكول TCP التحكم في الازدحام، وعرفنا أن التحكم في الازدحام ضروري لصحة الشبكة. بدون تحكم في الازدحام يمكن

بسهولة أن تصبح الشبكة مقفولة، بحيث يتم نقل القليل من البيانات أو لا يتم نقل أي بيانات على الإطلاق من طرف إلى طرف. وفي الجزء 3-7 عرفنا أيضاً أن بروتوكول TCP يطبّق آلية تحكم في الازدحام من طرف إلى طرف، حيث يتم من خلالها زيادة معدل الإرسال بشكلٍ خطي عند توقع كون مسار الإرسال خالياً من الازدحام، وتقليل معدل الإرسال بشكلٍ أسّي عند حدوث فقد في الرزم. كما تسعى تلك الآلية أيضاً لإعطاء كل توصيلة TCP تمر عبر وصلة مزدحمة حصةً متساويةً من حيّز التردد للوصلة. وتناولنا أيضاً بشيءٍ من التفصيل تأثير إنشاء توصيلات TCP والبداية البطيئة على التأخير، فلاحظنا أنه في العديد من السيناريوهات المهمة، يسهم إنشاء التوصيلة ومرحلة البداية البطيئة بشكلٍ ملحوظ في التأخير من طرف إلى طرف. نؤكّد مرة أخرى هنا على أنه بينما تطوّرت أساليب التحكم في الازدحام على مر السنين، فإنها تبقى مجالاً حيوياً للبحث ومن المحتمل أن يتواصل ذلك التطور خلال السنوات القادمة.

تركزت مناقشتنا للبروتوكولات المحددة لنقل البيانات على الإنترنت في هذا الفصل على بروتوكولي UDP وTCP – "خيول العمل" في طبقة النقل على الإنترنت. ومع ذلك فقد اتضح من خلال عقدين من التجربة مع هذين البروتوكولين الظروف التي لا يصلح فيها أيٌّ منهما بشكلٍ مثالي. ولذا فقد انكب الباحثون على تطوير بروتوكولات إضافية لطبقة النقل، أصبح عددٌ منها الآن معايير مقترحة لدى فريق عمل هندسة الإنترنت (IETF) نذكر منها ما يلي:

- بروتوكول التحكم في الازدحام لوحدة البيانات (DCCP) [RFC 4340]، والذي يوفر خدمةً للنقل غير الموثوق أساسها رسالة البيانات وبأعباء إضافية أقل. تشبه تلك الخدمة خدمة UDP ولكن بتحكم في الازدحام يختاره التطبيق ومتوائم مع بروتوكول TCP. إذا احتاج تطبيق ما نقلاً موثوقاً أو شبه موثوق للبيانات، يتم تحقيق ذلك ضمن التطبيق نفسه، ربما باستخدام الآليات التي درسناها في الجزء 3-4. يُتوقع استخدام بروتوكول DCCP بواسطة تطبيقات مثل العرض المستمر لمواد الوسائط المتعددة (انظر الفصل

السابع) التي يمكنها الاستفادة من الموازنة ما بين الوصول في الوقت المناسب والموثوقية في توصيل البيانات، ولكنها بحاجة أيضاً للتجاوب مع العواقب الوخيمة لازدحام الشبكة.

- بروتوكول النقل بالتحكم في مسارات البيانات (SCTP) [RFC 2960; RFC 3286]، وهو بروتوكول للنقل الموثوق أساسه رسالة البيانات ويسمح بتجميع أكثر من مسارٍ بيانات على مستوى التطبيقات عبر توصيلة SCTP واحدة (أسلوب يعرف باسم المسارات المتعددة). من وجهة نظر الموثوقية تعالَج المسارات المختلفة ضمن التوصيلة بشكلٍ منفصل كي لا يؤثر فقد الرزم في مسارٍ على توصيل البيانات في المسارات الأخرى. يسمح بروتوكول SCTP أيضاً بنقل البيانات على طريقين خارجين عندما يكون المضيف موصلاً بشبكتين أو أكثر، وبتوصيل اختياري للبيانات التي تصل بغير الترتيب السليم، بالإضافة إلى عدد من السمات الأخرى. يستخدم بروتوكول SCTP تقريباً نفس خوارزميات ضبط التدفق والتحكم في الازدحام التي يستخدمها بروتوكول TCP.
- بروتوكول التحكم في معدل البيانات المتوائم مع TCP (TFRC)، وهو بروتوكول للسيطرة على الازدحام وليس بروتوكولاً كاملاً لطبقة النقل [TFRC 2448]. يحدّد البروتوكول آليةً للسيطرة على الازدحام يمكن أن تستخدم في بروتوكول نقل آخر مثل DCCP (في الحقيقة فإن بروتوكول TFRC هو أحد البروتوكولين اللذين يمكن اختيارهما بواسطة التطبيق والمتوفرين في DCCP). إن الهدف من TFRC هو تنعيم نمط سن المنشار لسلوك التحكم في الازدحام في بروتوكول TCP (انظر الشكل 3-51)، وفي الوقت نفسه الحفاظ على معدل إرسال على المدى البعيد يقارب إلى حدٍ معقول معدل إرسال TCP. بمعدل إرسال أكثر سلاسة مما عليه الحال في TCP، يكون بروتوكول TFRC ملائماً لتطبيقات الوسائط المتعددة مثل

هاتف الإنترنت والنقل المستمر لمواد الوسائط المتعددة، حيث يكون من المهم استخدام معدل إرسال سَلِس. وجدير بالذكر أن TFRC هو بروتوكول مبني على معادلة، حيث تُستخدم النسبة المقاسة لفقد الرزم كمتغير في المعادلة [Padhye 2000] التي تقوم بتقدير الطاقة الإنتاجية لبروتوكول TCP إذا ما كانت جلسة TCP تعاني من فقد الرزم بتلك النسبة. يؤخذ هذا المعدل عندئذٍ كمعدل الإرسال المستهدف لبروتوكول TFRC.

ستُخبر الأيام عمّا إذا كانت البروتوكولات DCCP أو SCTP أو TFRC سترى انتشاراً على نطاق واسع. رغم أن تلك البروتوكولات توفر إمكانيات أفضل بشكلٍ واضح مقارنةً ببروتوكولي TCP وUDP، فإن هذين الأخيرين من ناحية أخرى قد أثبتا أنهما "جيدان بما فيه الكفاية" على مرّ السنين. هل سيفوز "الأفضل" على "الجيد بما فيه الكفاية"؟ سوف يعتمد ذلك على تركيبة معقدة من الاعتبارات التقنية والاجتماعية والتجارية.

ذكرنا في الفصل الأول أن شبكة الحاسب يمكن تقسيمها إلى "حافة الشبكة" و" قلب الشبكة". تغطي حافة الشبكة كل شيء يحدث في الأنظمة الطرفية. الآن وبعد أن غطينا طبقة التطبيقات وطبقة النقل نكون قد انتهينا من مناقشتنا لحافة الشبكة بالكامل. لقد حان الوقت الآن لاستكشاف قلب الشبكة! ستبدأ هذه الرحلة في الفصل القادم حيث سندرس طبقة الشبكة، وتستمر إلى الفصل الخامس حيث سندرس طبقة ربط البيانات.

أسئلة وتمارين وتدريبات الفصل الثالث

❖ أسـئلة مراجعة

• الأجزاء 3-1 إلى 3-3

1. افترض أن طبقة الشبكة توفر الخدمة التالية: تقبل طبقة الشبكة من المضيف المصدر قطعة بيانات بطول أقصى قدره 1200 بايت وعنوان وجهة من طبقة النقل. بعد ذلك تضمن طبقة الشبكة تسليم القطعة إلى طبقة النقل على مضيف الوجهة. افترض أن مضيف الوجهة يمكن أن يدعم العديد من عمليات تطبيقات الشبكة والتي يتم تنفيذها عليه في نفس الوقت.
   a. صمم أبسط بروتوكول ممكن لطبقة النقل والذي يقوم بنقل بيانات التطبيق إلى العملية على مضيف الوجهة. افترض أن نظام التشغيل على مضيف الوجهة قد خصّص رقم منفذ من 4 بايتات لكل عملية تطبيقات يجري تنفيذها على المضيف.
   b. قم بتعديل هذا البروتوكول بحيث يوفر "عنوان عودة" إلى عملية الوجهة.
   c. في بروتوكولاتك أعلاه، هل على طبقة التطبيقات أن تعمل شيئاً في قلب شبكة الحاسب؟
2. تخيّل كوكباً ينتمي كل شخص عليه لعائلة من 6 أفراد، وكل عائلة تعيش في بيتها الخاص بها، وكل بيت له عنوانه الفريد الخاص به. افترض أن هذا الكوكب تتوافر به خدمة بريدية لنقل البريد من منزل المصدر إلى منزل الوجهة. تتطلب خدمة البريد أن: (1) يكون الخطاب في ظرف، و(2) يكون عنوان منزل الوجهة (ولاشيء غيره) مكتوباً بوضوح على الظرف. افترض أن كل عائلة انتدبت أحد أفرادها لجمع وتوزيع البريد لأفراد العائلة الآخرين. لا تبيّن الخطابات بالضرورة أي دلالة على الشخص الموجّه له الخطاب.
   a. باستخدام الحل للتمرين 1 أعلاه، صف بروتوكولاً يمكن لمندوبي العائلات استخدامه لتسليم البريد من فرد مرسل في عائلة إلى فرد مستلم في عائلة.
   b. في بروتوكولك أعلاه، هل يحدث أبداً أن تضطر خدمة البريد لفتح ظرف وفض خطاب لكي تتمكن من تقديم خدمتها؟

3. خذ في الاعتبار توصيلة TCP بين المضيف A والمضيف B. افترض أن قطع TCP لها رقم منفذ المصدر x ورقم منفذ الوجهة y. ما رقم منفذ المصدر ومنفذ الوجهة للقطع التي تنتقل من المضيف B إلى المضيف A؟
4. اذكر لماذا قد يختار مطور تطبيق تشغيل تطبيقه على بروتوكول UDP بدلاً من بروتوكول TCP.
5. لماذا ترسل حركة بيانات الصوت والفيديو عادةً على بروتوكول TCP بدلاً من بروتوكول UDP في إنترنت اليوم؟ (ملاحظة: الإجابة التي نتوقعها ليس لها أي علاقة بآلية بروتوكول TCP للسيطرة على الازدحام).
6. هل يمكن لتطبيقٍ ما الاستمتاع بخدمة نقل موثوقة للبيانات رغم أن التطبيق يعمل بالفعل على بروتوكول UDP؟ إذا كان الأمر كذلك، فكيف؟
7. افترض أن عملية على المضيف ج لها مقبس UDP برقم المنفذ 6789. افترض أن كلاً من المضيفين A وB يرسل قطعة UDP إلى المضيف C برقم منفذ الوجهة يساوي 6789. هل سيتم توجيه كلتا القطعتين إلى نفس المقبس على المضيف C؟ إذا كان الأمر كذلك، كيف ستستطيع العملية على المضيف ج أن تعرف أن القطعتين نشأتا على مضيفين مختلفين؟
8. افترض أن خادم ويب يعمل على المضيف C على منفذ 80. افترض أن هذا المضيف يستخدم توصيلات دائمة، ويتلقى حالياً طلبات من مضيفين مختلفين، A وB. هل يتم إرسال كل الطلبات عبر نفس المقبس على المضيف C؟ إذا كانت الطلبات تمر عبر مقبسين مختلفين، هل كلا المقبسين لهما المنفذ 80؟ ناقش واشرح.

- **الجزء 3-4**

9. في بروتوكولات rdt التي طوّرناها، لماذا احتجنا إلى استخدام الأرقام التسلسلية؟
10. في بروتوكولات rdt التي طوّرناها، لماذا احتجنا إلى استخدام موقّتات؟
11. افترض أن زمن تأخير رحلة الذهاب والإياب بين المرسِل والمستقبل ثابت ومعروف للمرسل. هل سيكون من الضروري في بروتوكول rdt 3.0 استخدام موقّت، على افتراض أن الرزم يمكن أن تضيع؟ اشرح.
12. قم بزيارة برنامج جافا التفاعلي الخاص ببروتوكول العودة N للوراء على موقع الويب المصاحب لهذا الكتاب.

a. اجعل المصدر يرسل خمس رزم، ثم أوقف الرسوم التوضيحية المتحركة قبل وصول أي من تلك الرزم الخمس إلى الوجهة. بعد ذلك قم بإفناء أول رزمة ثم استأنف تشغيل الرسوم التوضيحية المتحركة. صف ما يحدث.

b. كرّر التجربة، ولكن الآن مع السماح بتدفق الرزمة الأولى إلى الوجهة وإفناء أول اشعار استلام. صف ما يحدث.

c. وأخيراً، حاول إرسال ست رزم. ماذا يحدث؟

13. كرّر تمرين 12، ولكن مع برنامج جافا التفاعلي الخاص ببروتوكول "الإعادة الانتقائية". ما هو الاختلاف بين بروتوكولي "ارجع N للوراء" و "الإعادة الانتقائية"؟

- **الجزء 3-5**

14. صح أم خطأ؟:

a. يقوم المضيف A بإرسال ملف كبير إلى المضيف B على توصيلة TCP. افترض أن المضيف B ليس لديه بيانات يريد إرسالها إلى المضيف A. لن يرسل المضيف B إشعارات استلام إلى المضيف A لأنه (أي المضيف B) لا يكون بوسعه تركيب إشعارات الاستلام على ظهر البيانات.

b. لا يتغير حجم نافذة الاستقبال RcvWindow في بروتوكول TCP أبداً طوال فترة التوصيلة.

c. افترض أن المضيف A يرسل إلى المضيف B ملفاً كبيراً على توصيلة TCP. لا يمكن أن يتجاوز عدد البايتات التي يرسلها A ولم يتم الإشعار باستلامها حجم مخزن الاستقبال المؤقت على B.

d. افترض أن المضيف A يرسل إلى المضيف B ملفاً كبيراً على توصيلة TCP. إذا كان الرقم التسلسلي لقطعة بيانات على هذه التوصيلة هو m، فإن الرقم التسلسلي لقطعة البيانات التالية هو بالضرورة $m+1$.

e. تتضمن قطعة بيانات TCP حقلاً في ترويستها يحتوي على حجم نافذة الاستقبال RcvWindow.

f. افترض أن قيمة آخر عينة SampleRTT على توصيلة TCP هي 1 ثانية. ستكون القيمة الحالية لفترة انقضاء الموقت TimeoutInterval بالضرورة أكبر من ثانية.

g. افترض أن المضيف A يرسل إلى المضيف B على توصيلة TCP قطعة بيانات تحمل 4 بايتات من البيانات ولها الرقم التسلسلي 38. في هذه القطعة يكون رقم إشعار الاستلام هو بالضرورة 42.

15. افترض أن المضيف A يرسل إلى المضيف B على توصيلة TCP قطعتي TCP الواحدة تلو الأخرى مباشرة. تحمل القطعة الأولى الرقم التسلسلي 90 بينما تحمل الثانية الرقم التسلسلي 110.
   a. كم بايت من البيانات تحمل القطعة الأولى؟
   b. افترض أن القطعة الأولى تُفقد في الطريق ولكن القطعة الثانية تصل إلى المضيف B. ماذا سيكون رقم إشعار الاستلام الذي يرسله المضيف B إلى المضيف A
16. خذ في الاعتبار مثال بروتوكول الوصول للحاسبات عن بعد (Telnet) الذي أوردناه في الجزء 3-5. بعد بضع ثوان من إدخال المستخدم للحرف 'C' يقوم بإدخال الحرف 'R'. بعد إدخال الحرف 'R'، كم عدد قطع البيانات التي يتم إرسالها؟ وما هي محتويات حقلي الرقم التسلسلي ورقم إشعار الاستلام على كل قطعة؟

- **الجزء 3-7**

17. افترض أن توصيلتي TCP موجودتان على وصلة عنق زجاجة لها معدل إرسال R بت/ثانية. لدى كلٍّ من التوصيلتين ملف ضخم مطلوب إرساله (في نفس الاتجاه على وصلة عنق الزجاجة). يبدأ إرسال الملفين في نفس الوقت. ما هو معدل الإرسال الذي يود بروتوكول TCP أن يعطيه لكلٍ من التوصيلتين؟
18. صح أم خطأ؟: خذ في الاعتبار السيطرة على الازدحام في بروتوكول TCP. عند انقضاء فترة الموقّت عند المرسل، يتم ضبط قيمة العتبة (Threshold) عند نصف قيمتها السابقة.

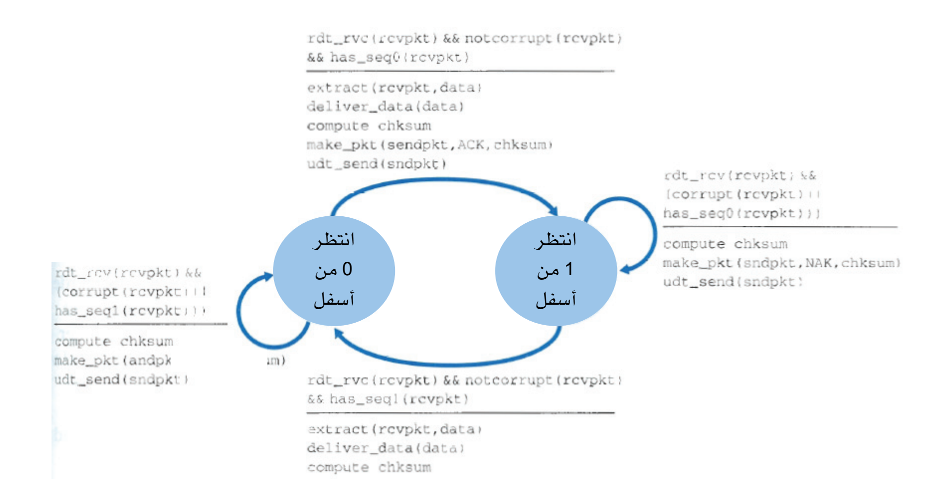
❖ تدريبات

1. افترض أن الزبون A يبدأ جلسة Telnet مع الخادم S، وفي الوقت نفسه تقريباً يُنشىء الزبون B جلسة Telnet مع نفس الخادم S. أوجد أرقام منافذ ممكنة للمصدر والوجهة لكلٍّ من:
   a. قطع البيانات المرسلة من A إلى S.
   b. قطع البيانات المرسلة من B إلى S.

c. قطع البيانات المرسلة من S إلى A.
d. قطع البيانات المرسلة من S إلى B.
e. إذا كان A وB مضيفين مستقلين، هل يمكن أن يكون رقم منفذ المصدر في القطع من A إلى S هو نفسه من B إلى S؟
f. ماذا لو كانا نفس المضيف؟

2. خذ في الاعتبار الشكل 3-5. ما قيمة منافذ المصدر والوجهة في قطع البيانات التي تتدفق من الخادم إلى عمليات الزبائن؟ ما هي عناوين IP في وحدات بيانات طبقة الشبكة التي تحمل قطع طبقة النقل؟
3. يستخدم كلٌّ من بروتوكولي برنامج UDP وTCP مكمّل الواحد لنظام المجموع التدقيقي. لنفترض أن لديك البايتات الثلاثة التالية: 01010101، 01110000، 01001100. ما هو مكمّل الواحد لمجموع تلك البايتات؟ (لاحظ أنه رغم أن UDP وTCP يستخدمان في الواقع كلمات (words) تتكون كلٌّ منها من 16 بتاً لحساب المجموع التدقيقي، فالمطلوب منك في هذا التمرين الحصول على المجموع لبايتات يضم كلٌّ منها 8 بتات فقط. وضّح كل الخطوات. لماذا يأخذ UDP مكمّل الواحد للمجموع، أي لماذا لا يأخذ المجموع نفسه وحسب؟ في نظام مكمّل الواحد، كيف يمكن للمستقبل اكتشاف الأخطاء؟ هل من الممكن مرور خطأ في بت واحد دون أن يُكتشف؟ ماذا عن خطأ في بتين؟
4. a. افترض أن لديك البايتين التاليين: 00110100 و01101001. ما هو مكمّل الواحد لمجموع هذين البايتين؟
b. افترض أن لديك البايتين التاليين: 11110101 و00101001. ما هو مكمّل الواحد لمجموع هذين البايتين؟
c. للبايتين في الجزء (a)، اعط مثالا لبت واحد لو انقلب (من 0 إلى 1 أو من 1 إلى 0) لما تغير مكمّل الواحد لمجموع البايتين.
5. افترض أن مستقبِل UDP يحسب مجموع الإنترنت التدقيقي لقطعة UDP التي يستقبلها ويجد أنه مطابق للقيمة الموجودة في حقل المجموع التدقيقي في القطعة. هل يمكن للمستقبِل أن يكون متأكداً تماماً بدون أدنى شك أنه لا توجد بتات خطأ في القطعة. اشرح إجابتك.
6. خذ في الاعتبار دوافعنا لتصحيح بروتوكول rdt2.1. وضّح أن المستقبل، والذي يظهر في الشكل على الصفحة التالية، عندما يعمل مع المرسل المبين في الشكل 3-11، فإنه يمكن أن يؤدي بالمرسِل والمستقبل للدخول في حالة جمود مستمر، حيث ينتظر كلٌّ منهما حدثاً لن يقع أبداً.

7. في بروتوكول rdt3.0 ، لا تحمل إشعارات الاستلام التي تتدفق من المستقبل إلى المرسل أرقاماً تسلسلية (رغم أنها تحمل الرقم التسلسلي للرزم التي تُشعر باستلامها). لماذا لا تحتاج إشعارات الاستلام التي نرسلها إلى أرقامٍ تسلسلية؟
8. ارسم آلة الأوضاع المحدودة (FSM) على جانب المستقبل من بروتوكول rdt3.0.
9. اعط تعقباً لتسلسل الأحداث لتشغيل بروتوكول rdt3.0 عندما تكون كلٌّ من رزم البيانات ورزم إشعار الاستلام عرضةً للأخطاء. ينبغي أن تشبه إجابتك تلك المستخدمة في الشكل 3-16.
10. خذ في الاعتبار قناة يمكن أن تفقد رزماً ولكن لها تأخير له حد أقصى معروف. قم بتعديل بروتوكول rdt2.1 ليشمل انقضاء وقت الموقّت وإعادة الإرسال على المرسل. وضّح كيف يمكن لبروتوكولك الاتصال بشكلٍ صحيح عبر تلك القناة.



11. يتجاهل جانب المرسل من بروتوكول rdt3.0 ببساطة (أي لا يتخذ أي إجراء بشأن) كل الرزم التي يتم استلامها وبها خطأ أو فيها خطأ في حقل رقم الإشعار ack-num في رزمة إشعار بالاستلام. افترض أنه في مثل هذه الظروف، يقوم rdt3.0 ببساطة بإعادة إرسال رزمة البيانات الحالية. فهل يظل البروتوكول يعمل؟ (ملاحظة: خذ في الاعتبار ما يمكن أن يحدث إذا لم يكن هناك سوى أخطاء في البتات فقط؛ ولا يوجد أي فقد في الرزم، ولكن يمكن حدوث انتهاء لفترة الموقت قبل الأوان. خذ في الاعتبار عدد المرات التي ستُرسل فيها الرزمة رقم n في نهاية الأمر عندما تقارب n من اللانهاية.

12. خذ في الاعتبار بروتوكول rdt3.0. وضح برسم بياني أنه إذا كان بوسع توصيلة الشبكة بين المرسِل والمستقبِل إعادة ترتيب الرسائل (أي إذا كان يمكنها تبديل ترتيب رسالتين من الرسائل التي تنتقل على الوسط مابين بين المرسِل والمستقبِل)، فإن بروتوكول البت المتناوب لن يعمل بشكلٍ صحيح (تأكد من تحديد كيف أنه لن يعمل بشكلٍ صحيح). في الرسم ينبغي أن يكون المرسِل على اليسار والمستقبِل على اليمين، ومحور الوقت يمتد من أعلى الصفحة إلى أسفلها، مع بيان تبادل البيانات (D) وإشعارات الاستلام (A). تأكد من توضيح الرقم التسلسلي المرتبط بكل رزمة بيانات أو إشعار استلام.
13. خذ في الاعتبار بروتوكولاً للنقل الموثوق للبيانات يستخدم فقط إشعارات استلام سلبية. افترض أن المرسِل يرسل بيانات فقط على فترات متباعدة. هل يكون استخدام بروتوكول يرسل إشعارات استلام سلبية أفضل من استخدام بروتوكول يرسل إشعارات استلام إيجابية؟ لماذا؟ افترض الآن أن المرسِل لديه الكثير من البيانات لإرسالها وأن التوصيلة من طرف إلى طرف تعاني من فقد قليل في البيانات. في تلك الحالة الثانية، هل يكون استخدام بروتوكول يرسل إشعارات استلام سلبية أفضل من استخدام بروتوكول يرسل إشعارات استلام إيجابية؟ لماذا؟
14. خذ في الاعتبار مثالاً لاتصال يمر عبر بلد كبير كالمبيّن في الشكل 3-17. ماذا ينبغي أن يكون حجم النافذة ليصبح مدى استغلال الشبكة أكثر من 90٪.
15. في بروتوكول الإعادة الانتقائية (SR) العام الذي درسناه في الجزء 3-4-4، يرسل المرسل الرسالة بمجرد توفرها (إذا كانت ضمن النافذة) وبدون انتظار وصول إشعار استلام. لنفترض الآن أننا نريد بروتوكول إعادة انتقائية يرسل الرسائل اثنتين في كل مرة. بمعنى أن المرسل يرسل زوجاً من الرسائل، وسوف يرسل الزوج التالي من الرسائل فقط عندما يعلم أن كلتا الرسالتين المرسلتين ضمن الزوج الأول قد وصلتا بشكلٍ صحيح. لنفترض أن القناة يمكن أن تفقد الرسائل ولكنها لا تفسدها ولا تعيد ترتيبها. قم بتصميم بروتوكول للتحكم في الخطأ أثناء النقل الموثوق للرسائل باتجاه واحد. بيّن آلة أوضاع محدودة (FSM) تعطي وصفاً للمرسل والمستقبِل. صِف صيغة الرزم المُرسَلة بين المرسل والمستقبِل، والعكس بالعكس. إذا استخدمت أياً من الإجراءات غير تلك المذكورة في الجزء 3-4 (على سبيل المثال udt_send() وstart_timer()، وrdt_rcv ()، وهلم جرّا)، فقم بتوضيح وظيفة كل إجراء. اعط مثالاً (جدولاً زمنياً لتعاقب الأحداث في المرسل و المستقبِل) يبيّن كيف يمكن لبروتوكولك التعافي من فقد رزمة.
16. خذ في الاعتبار سيناريو يريد فيه المضيف A إرسال رزم في نفس الوقت إلى كلٍّ من المضيف B والمضيف C. يتصل المضيف A بكلٍ من المضيف B وC عن طريق قناة إذاعة

– كل رزمة تُرسل من A تُنقل إلى كلٍّ من B وC على القناة. افترض أن قناة الإذاعة التي تربط مابين A وB وC يمكن أن تفقد أو تُفسد الرزم بشكلٍ مستقل (ومن ثم يمكن مثلاً لرزمة مرسلة من A أن تصل سليمة إلى B ولكن ليس إلى C). صمّم بروتوكولاً للتحكم في الخطأ من نوع "قف وانتظر" للنقل الموثوق للرزم من A إلى B وC بحيث لا يحصل A على بيانات جديدة من الطبقة الأعلى قبل التأكد من أن كلاً من B وC قد استلم الرزمة الحالية صحيحة. اعط وصفاً لآلة الأوضاع المحدودة (FSM) على كلٍّ من A وC. (ملاحظة: يجب أن تكون آلة الأوضاع المحدودة (FSM) على المضيف B هي نفسها على المضيف C تقريباً). قم أيضاً بإعطاء وصف لصيغ الرزم المستخدمة.

17. خذ في الاعتبار سيناريو يريد فيه كلٌّ من المضيف A و B إرسال رسائل الى C. المضيف A يرتبط بالمضيف C بقناة يمكن أن تُفقد وتُفسد (ولكن ليس إعادة ترتيب) الرسائل. يتصل المضيفان B وC بقناة أخرى (مستقلة عن القناة التي تربط A بـ C) لها نفس المواصفات. يجب أن تتناوب طبقة النقل الموجودة على C في تسليم الرسائل من A إلى B إلى الطبقة الأعلى (أي أنها ينبغي أن تسلّم أولاً البيانات من الرزمة من A، ثم البيانات من الرزمة من B، وهكذا). صمّم بروتوكولاً للتحكم في الخطأ من نوع قف وانتظر للنقل الموثوق للرزم من A إلى B إلى C بتسليم متبادل للبيانات عند C كما هو موضّح أعلاه. اعط وصفاً لآلة الأوضاع المحدودة (FSM) على كلٍّ من A و C. (ملاحظة: يجب أن تكون آلة الأوضاع المحدودة (FSM) على المضيف ب هي نفسها على المضيف A تقريباً). قم أيضاً بإعطاء وصف لصيغ الرزم المستخدمة.

18. خذ في الاعتبار بروتوكول "ارجع N للوراء" (GBN) فيه حجم نافذة المستقبل 3، ومدى الأرقام التسلسلية 1024. افترض أنه عند الوقت t يتوقع المستقبل استلام الرزمة ذات الرقم التسلسلي k كالرزمة التالية بالترتيب السليم. افترض أن الوسط لا يعيد ترتيب الرسائل. أجب على الأسئلة التالية:

a. ما هي مجموعات الأرقام التسلسلية الممكن وجودها داخل نافذة المرسل عند الوقت t؟ برّر إجابتك.

b. ما هي كل القيم الممكنة في حقل ACK في كل الرسائل الممكنة التي تنتقل حالياً عائدةً إلى المرسل عند الوقت t؟ برّر إجابتك.

19. افترض أن لدينا كيانان A وB على شبكة. يوجد لدى B مصدر بيانات سترسلها إلى A وفقاً للترتيبات التالية. عندما يتلقى A طلباً من الطبقة الأعلى لإحضار رسالة البيانات التالية (D) من B، يتعين على A إرسال رسالة طلب (R) إلى B على القناة من A إلى B. فقط عندما يتلقى B رسالة R يقوم بإرسال رسالة البيانات (D) إلى A على القناة من B إلى A. يجب على A تسليم نسخة واحدة فقط من كل رسالة بيانات (D) إلى الطبقة

الأعلى. يمكن أن تضيع الرسائل R (ولكن لا تفسد) على القناة من A إلى B؛ بمجرد إرسال الرسائل D فإنها تصل صحيحة دائماً. التأخير عبر كلا القناتين غير معروف ومتغير. صمّم (اعط وصفاً لآلة الأوضاع المحدودة (FSM)) بروتوكولاً يتضمن الآليات المناسبة للتعويض عن الفقد الذي تتعرض له البيانات على القناة من A إلى B ويقوم بتمرير الرسالة إلى الطبقة الأعلى في الكيان A كما هو موضّح أعلاه. استخدم فقط تلك الآليات الضرورية جداً.

20. خذ في الاعتبار بروتوكول العودة N للوراء (GBN) وبروتوكول الإعادة الانتقائية (SR). افترض أن مدى الأرقام التسلسلية هو k. ما هي أكبر نافذة مرسل مسموح بها من شأنها تجنّب حدوث مشاكل كالمبينة في الشكل 3-27 لكلٍ من هذين البروتوكولين؟

21. أجب بصح أو خطأ على كلٍّ من الأسئلة التالية وبرر إجابتك بإيجاز:

a. مع بروتوكول الإعادة الانتقائية (SR)، يمكن أن يتلقّي المرسل إشعار استلام (ACK) لرزمة تقع خارج نافذته الحالية.

b. مع بروتوكول العودة N للوراء (GBN)، يمكن أن يتلقّي المرسل إشعار استلام (ACK) لرزمة تقع خارج نافذته الحالية.

c. بروتوكول البت المتناوب هو نفسه بروتوكول الإعادة الانتقائية (SR) بنافذة إرسال ونافذة استقبال حجم كلٍّ منهما 1.

d. بروتوكول البت المتناوب هو نفسه بروتوكول العودة N للوراء (GBN) بنافذة إرسال ونافذة استقبال حجم كلٍّ منهما 1.

22. ذكرنا أن التطبيق يمكنه اختيار UDP كبروتوكول نقل لأن UDP يوفر تحكماً أدق (مقارنةً ببروتوكول TCP) للتطبيق في أي البيانات يتم إرسالها في قطعة البيانات ومتى يتم ذلك.

a. لماذا يتمتع التطبيق بتحكم أكبر في أي بيانات يتم إرسالها في قطعة البيانات؟

b. لماذا يتمتع التطبيق بتحكم أكبر في وقت إرسال قطعة البيانات؟

23. خذ في الاعتبار نقل ملف طوله L بايت في وجود أخطاء من المضيف A إلى المضيف B. افترض أن الحجم الأقصى للقطعة (MSS) هو 1460 بايت.

a. ما هي القيمة القصوى للطول L بحيث لا تُستنفد أرقام TCP التسلسلية؟ تذكّر أن حجم حقل الرقم التسلسلي في بروتوكول TCP هو 4 بايتات.

b. لقيمة L التي حصلت عليها في (a) أعلاه، احسب الوقت اللازم لإرسال الملف. افترض إضافة ترويسة نقل وشبكة وربط بيانات بطول كلي قدره 66 بايتاً إلى كل قطعة بيانات قبل إرسال الرزمة الناتجة على وصلة بمعدل إرسال قدره 10

ميجابت/ثانية. اهمل التحكم في التدفق والسيطرة على الازدحام، بحيث يقوم المضيف A بضخ قطع البيانات الواحدة تلو الأخرى مباشرةً وبشكلٍ مستمر.

24. يتصل المضيفان A وB عبر توصيلة TCP، وقد استلم المضيف B بالفعل كل البايتات التي أرسلها A حتى البايت 248. افترض أن المضيف A يرسل بعد ذلك قطعتين متعاقبتين إلى المضيف B. تتضمن القطعة الأولى والثانية 40 و60 بايتاً من البيانات، على الترتيب. الرقم التسلسلي للقطعة الأولى هو 249، ورقم منفذ المصدر هو 503، ورقم منفذ الوجهة هو 80. يقوم المضيف B بإرسال إشعار استلام كلما استلم قطعة بيانات من المضيف A.
    a. في القطعة الثانية المرسلة من A إلى B، ما هو الرقم التسلسلي، ورقم منفذ المصدر، ورقم منفذ الوجهة؟
    b. إذا وصلت القطعة الأولى قبل القطعة الثانية، فما هو رقم إشعار الاستلام، ورقم منفذ المصدر، ورقم منفذ الوجهة في إشعار استلام القطعة التي تصل أولاً؟
    c. إذا وصلت القطعة الثانية قبل القطعة الأولى، فما هو رقم إشعار الاستلام في إشعار استلام القطعة التي تصل أولاً؟
    d. افترض أن القطعتين اللتين بعث بهما A وصلتا بنفس الترتيب إلى B. افترض أن إشعار الاستلام الأول فُقد في الطريق والثاني وصل بعد أول انقضاء لفترة الموقّت. ارسم مخططاً زمنياً يبيّن قطع البيانات تلك وكل القطع وإشعارات الاستلام الأخرى التي يتم إرسالها (افترض أنه لا يوجد أي فقد إضافي في الرزم). لكل قطعة بيانات في الشكل الذي سترسمه، بيّن الرقم التسلسلي وعدد بايتات البيانات؛ ولكل إشعار استلام تضيفه بيّن رقم إشعار الاستلام.

25. المضيفان A وB مرتبطان مباشرةً عن طريق وصلة بسرعة إرسال قدرها 200 ميجابت/ثانية. توجد توصيلة TCP بين المضيفين، ويقوم المضيف A بإرسال ملف بأخطاء إلى المضيف B عبر تلك التوصيلة. يمكن للمضيف A إرسال بيانات التطبيقات على الوصلة بسرعة 100 ميجابايت/ثانية ولكن المضيف B يمكنه قراءة مخزن الاستقبال المؤقت على TCP لديه بمعدّل أقصاه 50 ميجابت/ثانية. صف تأثير تحكم TCP في التدفق.

26. تم مناقشة كوكيز SYN في الجزء 3-5-6.
    a. لماذا يكون من الضروري لخادم استخدام رقم تسلسلي أولي خاص في SYNACK؟
    b. افترض أن المهاجم يعرف أن المضيف المستهدَف يستخدم كوكيز SYN. هل يمكن للمهاجم إنشاء توصيلات نصف مفتوحة أو مفتوحة تماما ببساطة عن طريق إرسال رزمة إشعار استلام ACK إلى المضيف المستهدَف؟ برر إجابتك.

27. خذ في الاعتبار إجراء TCP المستخدم لتقدير زمن رحلة الذهاب والإياب (*RTT*). افترض أن $\alpha = 0.1$ واعتبر أن $SampleRTT_1$ هي آخر (أحدث) عينة *RTT*، وأن $SampleRTT_2$ هي عينة *RTT* قبل الأخيرة، وهلم جرا...
   a. لتوصيلة TCP بعينها، افترض أنه تم إعادة أربعة إشعارات استلام بعينات *RTT* التالية: $SampleRTT_1$ و $SampleRTT_2$ و $SampleRTT_3$ و $SampleRTT_4$. عبّر عن القيمة المقدّرة لـ *RTT* (أي EstimatedRTT) بدلالة قيم عينات *RTT* الأربع.
   b. قم بتعميم النتيجة أعلاه لعدد *n* من عينات *RTT*.
   c. في المعادلة في الجزء (b)، دع *n* تؤول إلى ما لانهاية. علّق على سبب تسمية هذا الإجراء بالمتوسط المتحرّك الأسي.
28. في الجزء 3-5-3 ناقشنا تقدير *RTT* في بروتوكول TCP. لماذا في اعتقادك يتجنب بروتوكول TCP قياس SampleRTT لقطع البيانات المعاد إرسالها؟
29. ما هي العلاقة بين المتغير SendBase في الجزء 3-5-4 والمتغير LastByteRcvd في الجزء 3-5-5 ؟
30. ما هي العلاقة بين المتغير LastByteRcvd في الجزء 3-5-5 والمتغير *y* في الجزء 3-5-4؟
31. في الجزء 3-5-4 رأينا أن بروتوكول TCP ينتظر حتى يتلقّى ثلاثة إشعارات استلام (ACKs) مكرّرة قبل القيام بعملية إعادة إرسال سريعة (Fast Retransmit). لماذا في رأيك اختار مصممو بروتوكول TCP عدم القيام بإعادة إرسال سريعة بمجرد استلام أول إشعار استلام مكرر لقطعة بيانات؟
32. خذ في الاعتبار الشكل 3-46 (b). إذا زادت λ'_{in} عن $R/2$، هل يمكن أن تزيد λ_{out} عن $R/3$؟ اشرح. الآن خذ في الاعتبار الشكل 3-46 (c). إذا زادت λ'_{in} عن $R/2$، هل يمكن أن تزيد λ_{out} عن $R/4$ على افتراض أن الرزمة ستُمرّر مرتين في المتوسط من الموجّه إلى المستقبل؟ اشرح.
33. خذ في الاعتبار المخطط البياني التالي الذي يبيّن تغير حجم نافذة TCP كدالة في الوقت. افترض أن بروتوكول TCP رينو هو البروتوكول الذي يتعرّض للسلوك المبين أعلاه، أجب على الأسئلة التالية. وفي جميع الحالات، ينبغي توفير مناقشة قصيرة لتبرير إجابتك.
   a. حدّد الفترات من الوقت التي تكون فيها بداية TCP البطيئة شغّالة.
   b. حدّد الفترات من الوقت التي تكون فيها آلية TCP لتجنب الازدحام شغّالة.
   c. بعد جولة الإرسال رقم 16، هل يتم اكتشاف فقد قطعة عن طريق ثلاثة إشعارات استلام مكررة أم بانقضاء فترة الموقّت؟

d. بعد جولة الإرسال رقم 22، هل يتم اكتشاف فقد قطعة عن طريق ثلاثة إشعارات استلام مكررة أم بانقضاء فترة الموقّت؟

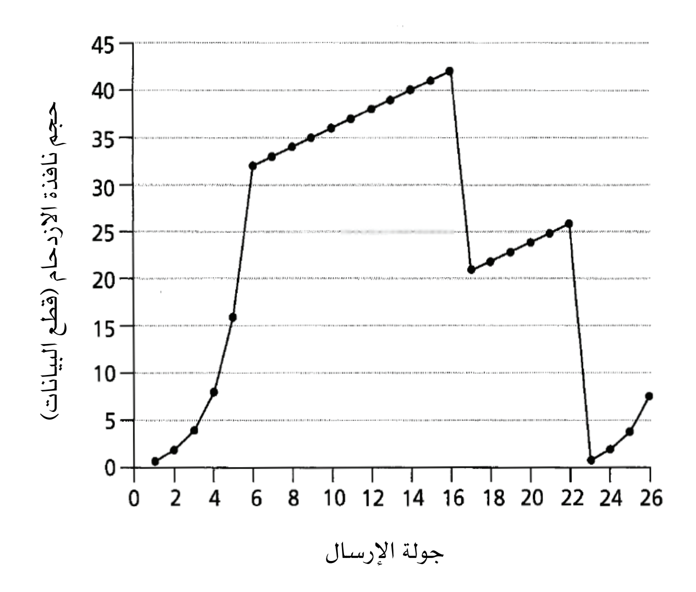
e. ما القيمة الأولية للعتبة (threshold) عند أول جولة إرسال؟

f. ما قيمة العتبة عند جولة الإرسال رقم 18؟

g. ما قيمة العتبة عند جولة الإرسال رقم 24؟

h. في أي جولة إرسال يتم إرسال قطعة البيانات رقم 70؟

i. بافتراض أنه يتم اكتشاف فقد رزمة بعد جولة الإرسال رقم 26 عن طريق تلقّي 3 إشعارات استلام مكرّرة. ماذا ستكون قيم كلٍّ من حجم نافذة الازدحام والعتبة؟



34. راجع الشكل 3-55، والذي يوضح تقارب خوارزمية AIMD لبروتوكول TCP. افترض أنه بدلاً من التناقص الضربي، يقوم TCP بتقليل حجم النافذة بكمية ثابتة. هل تؤول خوارزمية AIMD الناتجة في هذه الحالة إلى خوارزمية حصص متساوية؟. قم بتبرير إجابتك باستخدام مخطط بياني يماثل الشكل 3-55.

35. ناقشنا في الجزء 3-5-4 مضاعفة فترة انقضاء الموقّت بعد حدوث انقضاء للفترة. تُعتبر تلك الآلية شكلاً من أشكال السيطرة على الازدحام. لماذا يحتاج بروتوكول TCP الى آلية مبنية على مفهوم النافذة للسيطرة على الازدحام (كالتي درسناها في الجزء 3-7) بالإضافة إلى آلية مضاعفة فترة انقضاء الموقّت تلك؟

36. يقوم المضيف A بإرسال ملف ضخم بأخطاء عبر توصيلة TCP إلى المضيف ب. عبر تلك التوصيلة لا يوجد أي فقد للرزم، والموقتات لا تنقضي فتراتها أبداً. ارمز لمعدل الإرسال على الوصلة التي تربط المضيف A بالإنترنت بالرمز R بت/ثانية. افترض أن العملية في المضيف A بوسعها إرسال البيانات إلى مقبس TCP الخاص بها بمعدل S بت/ثانية، حيث $S = 10R$. افترض أيضاً أن مخزن الاستقبال المؤقت لبروتوكول TCP من الضخامة بحيث يمكن أن يسع الملف بأكمله، بينما يسع مخزن الإرسال المؤقت 1٪ من الملف. ما الذي يحول دون تمكّن العملية في المضيف A من تمرير البيانات باستمرار إلى مقبس TCP بمعدّل S بت/ثانية: تحكم TCP في التدفق؟ سيطرة TCP على الازدحام؟ أم شيء آخر؟ أجب بالتفصيل.

37. خذ في الاعتبار عملية إرسال ملف كبير من مضيف إلى آخر على توصيلة TCP لا يتم عليها أي فقد.

a. افترض أن TCP يستخدم خوارزمية AIMD للسيطرة على الازدحام دون بداية بطيئة. بافتراض أن CongWin تزداد بـ MSS واحدة في كل مرة تصل فيها دفعة إشعارات استلام (ACKs) وأن أوقات رحلة الذهاب والعودة (RTT) ثابتة تقريباً، كم نحتاج من الوقت لكي تزداد CongWin من 1MSS إلى 6MSS (على افتراض عدم حدوث فقد)؟

b. ما هو المتوسط العام (بدلالة MSS وRTT) لتلك التوصيلة حتى الوقت = $5RTT$؟

38. تذكّر الوصف الماكروسكوبي للطاقة الإنتاجية لبروتوكول TCP. في الفترة التي تغيّر فيها معدّل الإرسال على التوصيلة من $W/(2RTT)$ إلى W/RTT، فُقِدت رزمة واحدة (في آخر تلك الفترة تماماً).

c. بيّن أن معدل فقد الرزم (الكسر الذي يمثل الرزم المفقودة) هو:

$$L = \text{Loss rate} = \frac{1}{\frac{3}{8}W^2 + \frac{3}{4}W}$$

d. استخدم النتيجة أعلاه لتبيّن أنه إذا كان معدل الفقد على توصيلة هو L، فإن معدل الإرسال عليها يكون تقريباً:

$$\approx \frac{1.22\ \text{MSS}}{RTT\sqrt{L}}$$

39. في مناقشتنا لسمات بروتوكول TCP في الجزء 3-7، لاحظنا أنه لتحقيق طاقة إنتاجية قدرها 10 جيجا بت/ثانية يمكن للبروتوكول السماح باحتمال فقدان قطع بيانات قدره 2×10^{-10} (أي ما يعادل فقد قطعة واحدة من كل 5000000000 قطعة). بيّن اشتقاق

الاحتمال 2×10^{-10} (واحد لكل 5000000000) لقيم RTT وMSS المعطاة في الجزء 3-7. إذا كان على TCP توفير توصيلة بطاقة إنتاجية قدرها 100 جيجابايت/ ثانية، فما هو احتمال الفقد الأقصى المسموح به في القطع؟

40. في مناقشتنا للسيطرة على الازدحام في بروتوكول TCP في الجزء 3-7، افترضنا ضمناً أن مرسل TCP كان دائماً لديه بيانات يريد إرسالها. خذ في الاعتبار الآن الحالة التي يرسل فيها المرسِل كمية كبيرة من البيانات وبعد ذلك يتوقف عن الإرسال عند الوقت t_1 (لأنه لم يبق لديه بيانات لإرسالها). يبقى المرسِل عاطلاً لمدة طويلة نسبياً من الزمن ثم يستأنف بعدها إرسال المزيد من البيانات من جديد عند الوقت t_2، . ما هي مزايا وعيوب استخدام TCP لقيم CongWin وThreshold المستخدمة عند t_1 عند استئناف إرسال البيانات من جديد عند t_2؟ ما هو البديل الذي تقترحه؟ ولماذا؟

41. في هذا التمرين نبحث ما إذا كان أياً من بروتوكولي UDP أو TCP يوفر قدراً من توثيق النقطة الطرفية.

a. خذ في الاعتبار خادماً يتلقى طلباً ضمن رزمة UDP ويستجيب لهذا الطلب ضمن رزمة UDP (مثلاً على النحو المتّبع في خادم بنظام أسماء النطاقات DNS). إذا قام زبون له عنوان IP قيمته X بانتحال العنوان Y، فإلى أي عنوان سيرسل الخادم رده على الطلب؟

b. افترض الآن أننا نستخدم TCP وليس UDP، هل سيكون بوسع الزبون خداع الخادم لجعله يرسل ردّه إلى العنوان Y ببساطة بمجرد انتحال عنوان IP الخاص به؟

c. افترض أن خادماً يتلقى SYN مع عنوان مصدر IP قيمته Y، وبعد الاستجابة بـ SYNACK يستلم ويرسل إشعار استلام ACK على عنوان مصدر IP قيمته Y باستخدام الرقم الصحيح لإشعار الاستلام. بافتراض أن الخادم يختار الرقم التسلسلي الأوليّ بشكلٍ عشوائي وأنه لا يوجد هجوم من نوع "رجل في الوسط"، هل يمكن للخادم أن يكون على يقين من أن الزبون هو في الواقع Y (وليس زبوناً غيره له عنوان مختلف قام بانتحال العنوان Y)؟

42. في هذا التمرين سنأخذ في الاعتبار التأخير الذي ينشأ عن مرحلة البداية البطيئة في بروتوكول TCP. خذ في الاعتبار زبوناً وخادم ويب موصّلان عن طريق وصلة واحدة لها معدل إرسال R بت/ثانية. افترض أن الزبون يريد الحصول على كائن حجمه بالضبط يساوي $15S$، حيث S هو الحد الأقصى لحجم قطعة البيانات (MSS). افترض أن زمن رحلة الذهاب والإياب بين الزبون والخادم هو RTT (وافترض أنه ثابت). بإهمال ترويسات

البروتوكولات، احسب الوقت اللازم لكي يحصل الخادم على الكائن المطلوب (بما في ذلك وقت إنشاء توصيلة TCP) في كلٍّ من الحالات التالية:

a. $4S/R > S/R + RTT > 2S/R$
b. $8S/R > S/R + RTT > 4S/R$
c. $S/R > RTT$

❖ أسئلة للمناقشة

1. ما المقصود باختطاف توصيلة TCP ؟ كيف يمكن القيام بذلك؟
2. في الجزء 3-7 لاحظنا أن تطبيق زبون – خادم يمكنه بطريقة "غير عادلة" إنشاء العديد من توصيلات متوازية في وقت واحد. ما الذي يمكن عمله لجعل الإنترنت عادلة حقا؟
3. اقرأ البحوث المنشورة لمعرفة ما المقصود بالتعبير "متوائم مع TCP (TCP Friendly). أكتب وصفاً من صفحة واحدة للتواؤم مع TCP.
4. في نهاية الجزء 3-7-1 ناقشنا حقيقة أنه بوسع التطبيق فتح عدة توصيلات TCP للحصول على طاقة إنتاجية أعلى (أو بالمكافئ معدلات أعلى لنقل البيانات). ماذا يحدث لو أن كل التطبيقات حاولت تحسين أدائها عن طريق استخدام وصلات متعددة؟ ما هي بعض الصعوبات التي تكتنف محاولة عنصر من عناصر الشبكة تحديد ما إذا كان تطبيقٌ ما يستخدم وصلات TCP متعددة؟
5. بالإضافة إلى مسح منافذ TCP و UDP، ما هي الوظائف الأخرى لـ nmap؟ اجمع آثار رزم باستخدام إيثيريل (أو أي أداة أخرى لالتقاط الرزم) لعمليات تبادل رزم nmap. استخدم تلك الآثار لشرح كيف تعمل بعض السمات المتقدمة.
6. اقرأ البحوث المنشورة فيما يتعلق ببروتوكول النقل بالتحكم في مسارات البيانات (SCTP) [RFC 2960; RFC 3286]. ما هي التطبيقات التي يتصور مصممو SCTP أن يُستخدم فيها البروتوكول الجديد؟ ما هي السمات التي أضيفت إلى بروتوكول SCTP لتلبية احتياجات تلك التطبيقات؟

❖ أسئلة برمجة: تنفيذ بروتوكول للنقل الموثوق للبيانات

في تمرين مختبر البرمجة هذا، ستقوم بكتابة كود الإرسال والاستقبال على مستوى النقل لتنفيذ بروتوكول بسيط للنقل الموثوق للبيانات. هناك إصداران من هذا المختبر، إصدار خاص ببروتوكول البت المتناوب والآخر ببروتوكول العودة N إلى الوراء GBN. هذا المختبر سيكون ممتعاً – كما أن البروتوكول الذي ستطوره في نهاية المختبر لن يختلف كثيرا عما هو مطلوب في العالم الحقيقي.

نظراً لأنه قد لا يتوافر لديك أجهزة قائمة بذاتها (بنظام تشغيل يمكنك تعديله)، فإن الكود الذي ستطوّره ينبغي تشغيله من خلال بيئة محاكاة من البرامجيات والعتاد. ومع ذلك، فإن واجهة البرمجة التي يتم توفيرها لبرامجك – أي أجزاء الكود التي سوف تستدعي برامجك من أعلى ومن أسفل – تعتبر قريبة جداً مما يحدث في بيئة يونيكس حقيقية. (وبالفعل، فإن واجهات البرمجيات البينية الموصوفة في تدريب البرمجة هذا هي أكثر واقعية بكثير من المرسل والمستقبل بدورة تنفيذ لانهائية (Infinite Loop) والتي تستخدمها العديد من الكتب الدراسية). يتم أيضاً محاكاة بدء وإيقاف المؤقّتات، كما أن المقاطعة بسبب المؤقّت سوف تقوم بتفعيل برامجك الخاصة بالتعامل مع المؤقتات.

يوجد تدريب المختبر كاملاً، بالإضافة إلى الكود التي ستحتاج لتجميعها مع الكود الخاصة بك، على الموقع الخاص بهذا على الإنترنت: http://www.aw1.com/kurose-ross.

❖ مختبر إيثيريل: استكشاف بروتوكول TCP

في هذا المختبر سوف تستخدم متصفح الشبكة لديك للوصول إلى ملف من خادم ويب. كما هو الحال في مختبرات إيثيريل السابقة، ستستخدم إيثيريل لالتقاط الرزم الواصلة إلى جهاز الحاسب الخاص بك. ولكن خلافاً للمختبرات السابقة، سيكون بوسعك أيضاً تنزيل أثر رزم يمكن قراءتها بواسطة برنامج إيثيريل من خادم الويب الذي قمت بتنزيل ذلك الملف منه. في أثر الرزم ذلك من الخادم، ستجد الرزم التي تم إنشاؤها على الخادم نتيجة وصولك إلى خادم الويب. ستقوم بتحليل آثار الرزم على كلٍّ من جانبي الزبون والخادم لاستكشاف جوانب من عمل بروتوكول TCP. على وجه الخصوص، ستقوم بتقييم أداء توصيلة TCP بين حاسبك وخادم الويب. سوف تقوم بتتبع أداء نافذة TCP، واستقراء المعلومات عن فقدان الرزم، وإعادة الإرسال، و التحكم في التدفق، والسيطرة على الازدحام، وتقدير زمن رحلة الذهاب والإياب.

كما هو الحال مع جميع مختبرات إيثيريل، يوجد وصفٌ كاملٌ لهذا المختبر على الموقع الخاص بهذا الكتاب على الإنترنت: http://www.aw1.com/kurose-ross.

❖ مختبر إيثيريل: استكشاف بروتوكول UDP

في هذا المختبر القصير ستقوم بعمليات اقتناص وتحليل الرزم لتطبيقاتك المفضلة التي تستخدم بروتوكول UDP (كنظام أسماء النطاقات DNS و تطبيقات الوسائط المتعددة مثل Skype). كما عرفنا في الجزء 3-3، فإن UDP بروتوكول نقل بسيط بلا رتوش. في هذا المختبر سوف تدرس الحقول المختلفة لترويسة قطع بيانات UDP بالإضافة إلى حساب المجموع التدقيقي.

كما هو الحال مع جميع مختبرات إيثيريل، يوجد وصفٌ كاملٌ لهذا المختبر على الموقع الخاص بهذا الكتاب على الإنترنت: http://www.aw1.com/kurose-ross.

طبقة الشبكـة

The Network Layer

محتويات الفصل:

- مقدمة
- شبكات الدائرة الافتراضية وشبكات وحدات البيانات
- ماذا بداخل الموجّه؟
- بروتوكول الإنترنت (IP): التمرير والعنونة في الإنترنت
- خوارزميات التوجيه
- التوجيه في شبكة الإنترنت
- توجيه البث الإذاعي (العام) والتوجيه المتعدد (الجماعي)
- الخلاصة

عرفنا في الفصل السابق أن طبقة النقل توفر أشكالاً مختلفة للاتصال من عملية إلى عملية بالاعتماد على خدمة طبقة الشبكة للاتصال من مضيف إلى مضيف. وتعلمنا أيضاً أن طبقة النقل تؤدي هذا الدور دون أية معرفة عن كيفية تحقيق طبقة الشبكة في واقع الأمر لهذه الخدمة. لذا ربما تتساءل الآن: ما الذي تحت قلنسوة خدمة الاتصال من مضيف إلى مضيف، وما الذي يجعلها تحدث؟ وكيف تتم؟

في هذا الفصل سوف نتعلم كيف تحقق طبقة الشبكة بالضبط خدمة الاتصال من مضيف إلى مضيف. وسوف نرى أنه على خلاف طبقة النقل يوجد جزء من طبقة الشبكة في كل مضيف وموجّه في الشبكة. ولهذا السبب فإن بروتوكولات طبقة الشبكة من بين البروتوكولات الأكثر تحدّياً في رصّة البروتوكولات (ولذا فقد حظيت باهتمام كبير!).

وطبقة الشبكة أيضاً هي إحدى الطبقات الأكثر تعقيداً في رصّة البروتوكولات، ولذا فسيكون لدينا الكثير من الموضوعات للتغطية هنا. سنبدأ دراستنا بنظرة عامة عن طبقة الشبكة والخدمات التي يمكن أن توفرها. ثم سنعاود التعرض مرةً أخرى للطريقتين الرئيستين لهيكلة طبقة الشبكة لتوصيل الرزم: نموذج إرسال وحدات البيانات (datagram model) ونموذج الدائرة الافتراضية (virtual circuit model)، واللتين سبق أن تناولناهما لأول مرة في الفصل الأول. وسنرى الدور الأساسي الذي تلعبه "العنونة" (addressing) في توصيل الرزمة إلى مضيف الوجهة.

وسنجري في هذا الفصل تمييزاً مهماً بين وظيفة التمرير (forwarding) ووظيفة التوجيه (routing) لطبقة الشبكة. حيث يتضمّن "التمرير" نقل الرزمة من وصلة داخلة إلى وصلة خارجة ضمن "موجّه" واحد؛ بينما يشمل التوجيه كل موجّهات الشبكة والتي تحدد تفاعلاتها الجماعية - عن طريق بروتوكولات التوجيه - المسارات التي تأخذها الرزم أثناء رحلتها من المصدر إلى الوجهة. وسوف يساعدك

تذكر هذا التمييز (الفرق) كلما تقدمت خلال هذا الفصل على وضع العديد من الموضوعات التي سنتناولها في سياق ملائم.

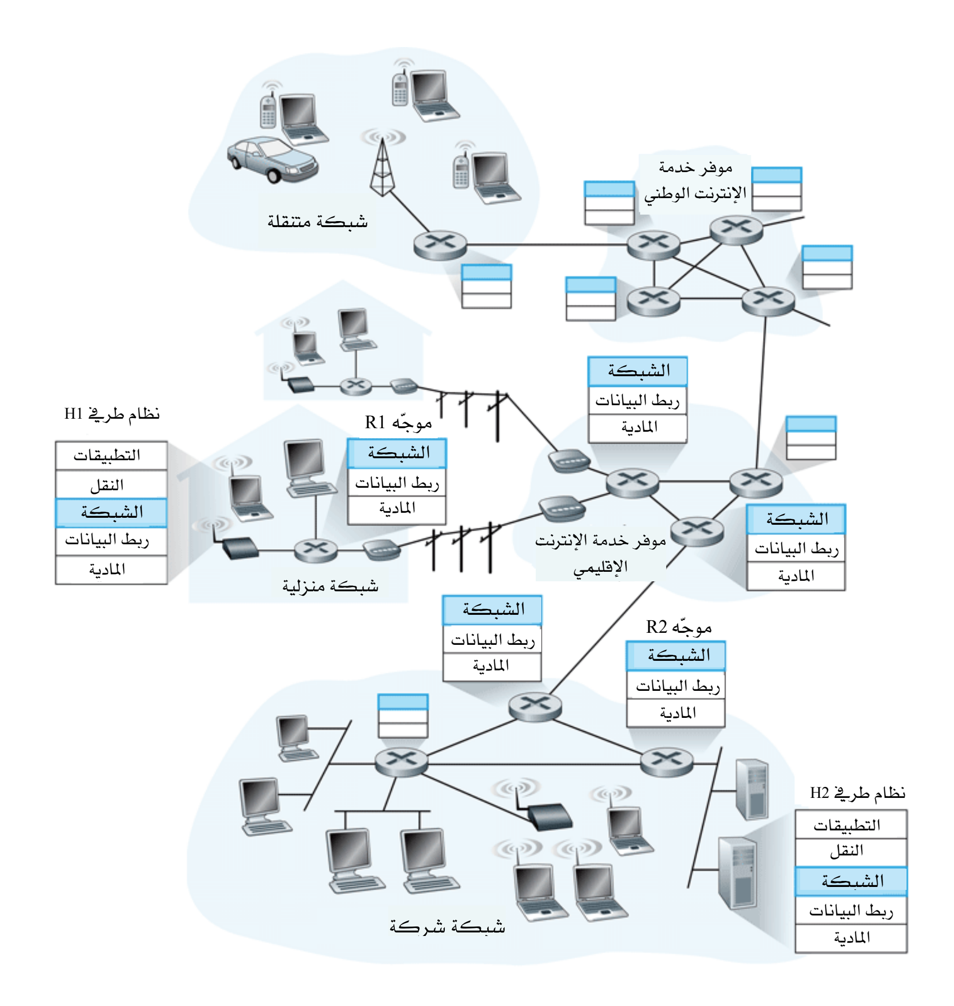
ولكي نُعمّق فهمنا لتمرير الرزم سننظر "داخل" موجّه لنرى بنيته المعمارية وتنظيم مكوناته المادية. ثم ننظر إلى تمرير الرزم في الإنترنت سوية مع بروتوكول الإنترنت الشهير IP. وسنفحص العنونة في طبقة الشبكة وصيغة رزمة بيانات الإصدار الرابع لبروتوكول الإنترنت (IPv4). وسوف ندرس أيضاً ترجمة عناوين الشبكة (NAT)، وتجزئة رزم البيانات (fragmentation)، وبروتوكول رسائل التحكم في الإنترنت (ICMP)، وبروتوكول IPv6.

بعد ذلك سنحوّل انتباهنا إلى وظيفة التوجيه في طبقة الشبكة، حيث سنرى أن مهمة خوارزمية التوجيه هي تحديد مسارات (طرق) جيدة من مصادر البيانات إلى وجهاتها. وسندرس أولاً نظرية خوارزميات التوجيه مع التركيز على النوعين الأكثر انتشاراً من الخوارزميات: خوارزمية "حالة الوصلة" (link state) وخوارزمية "متجه المسافة" (distance vector). ولأن خوارزميات التوجيه تزداد تعقيداً إلى حدٍّ كبير كلما زاد عدد الموجّهات في الشبكة، فإن أساليب التوجيه الهرمي (hierarchical routing) ستكون أيضاً محل اهتمامنا. وسنرى كيف توضع النظرية موضع التطبيق عندما نغطي بروتوكولات التوجيه داخل النظم المستقلة ذاتياً (intra-autonomous systems) في الإنترنت (مثل RIP وOSPF وIS-IS) وكذلك بروتوكول BGP للتوجيه بين النظم المستقلة ذاتياً (inter-autonomous systems) بها. وسننهي هذا الفصل بمناقشة توجيه البث الإذاعي (broadcast) والإرسال المتعدّد (multicast).

وباختصار يتكون هذا الفصل من ثلاثة أقسام رئيسة: حيث يغطي القسم الأول (الجزءان 4-1 و4-2) وظائف وخدمات طبقة الشبكة، ويغطي القسم الثاني (الجزءان 4-3 و4-4) التمرير، وأخيراً يغطي القسم الثالث (من الجزء 4-5 إلى الجزء 4-7) التوجيه.

4-1 مقدمة

يوضح الشكل 4-1 شبكة بسيطة ذات مضيفين H1 وH2، وعدّة موجّهات على المسار بين H1 وH2. افترض أن H1 يرسل معلومات إلى H2، ولننظر إلى دور طبقة الشبكة في كلا المضيفين وفي الموجّهات المتوسطة بينهما. تأخذ طبقة



الشكل 4-1 طبقة الشبكة.

الشبكة في H1 "قطع البيانات" (segments) من طبقة النقل في H1 ، وتغلف كل قطعة في "رزمة بيانات" (packet)، ثم بعد ذلك ترسل تلك الرزم إلى الموجّه المجاور R1. في مضيف الاستقبال H2، تستلم طبقة الشبكة رزم البيانات من الموجّه المجاور R2، وتستخرج "قطع بيانات" طبقة النقل، ثم تسلم تلك القطع إلى طبقة النقل في H2. إن الدور الأساسي للموجّهات هو إرسال رزم البيانات من الوصلات الداخلة الى الوصلات الخارجة. لاحظ أن رصّة البروتوكولات على الموجّهات في الشكل 4-1 مقطوعة (أى بدون طبقات عليا فوق طبقة الشبكة)، وذلك لأن الموجّهات لا تستخدم بروتوكولات طبقتي النقل والتطبيق كالتي درسناها في الفصلين الثانى والثالث (إلا لأغراض التحكم).

4-1-1 التمرير والتوجيه

هكذا قد يبدو – بشكلٍ خادع – أن دور طبقة الشبكة بسيط، فهي تنقل الرزم من مضيف الإرسال إلى مضيف الاستقبال فقط. يمكننا من البداية التعرف على وظيفتين مهمتين لطبقة الشبكة لتتمكن من أداء هذا الدور:

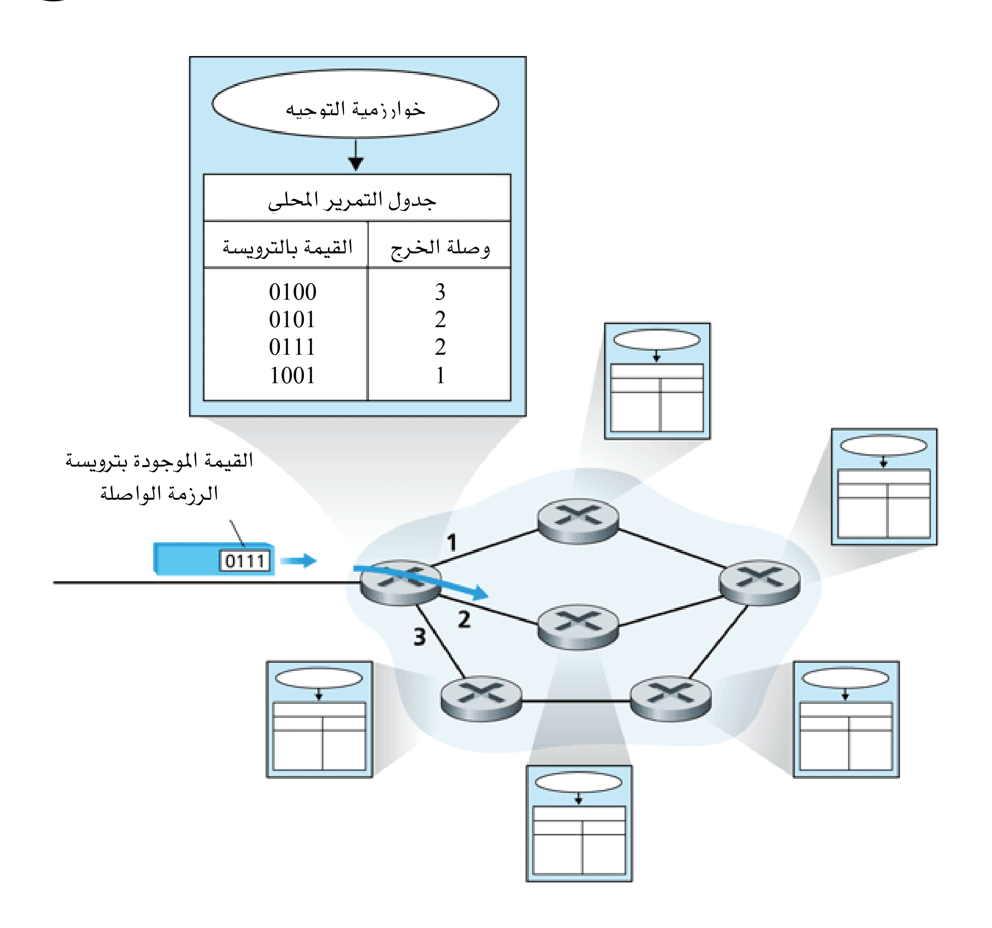
- التمرير: عندما تصل رزمة إلى وصلة مدخل (input link) للموجّه يجب عليه أن ينقل تلك الرزمة إلى وصلة مخرج (output link) مناسبة. على سبيل المثال عند وصول رزمة من المضيف H1 إلى الموجّه R1 يجب عليه أن يرسلها إلى الموجّه التالي على المسار إلى H2. في الجزء 4-3 سننظر داخل موجّه لنرى كيف تُرسل رزمة في الواقع من وصلة مدخل إلى وصلة مخرج على موجّه.
- التوجيه: يجب أن تقرّر طبقة الشبكة المسار الذي تتبعه الرزم وهي تتدفّق من المُرسِل إلى المستقبل. ويطلق على الخوارزميات التي تحسب هذه المسارات "خوارزميات التوجيه". على سبيل المثال ستحدد خوارزمية التوجيه الطريق الذي تتدفق خلاله الرزم من H1 إلى H2.

غالباً ما يُستخدم المصطلحان "تمرير" و"توجيه" بمعنى واحد من قِبَل بعض المؤلفين أثناء مناقشة طبقة الشبكة. لكننا في هذا الكتاب سوف نستخدمهما بدقة أكثر. يشير "التمرير" إلى عمل الموجّه المحلي لنقل رزمة من واجهة وصلة مدخل

(input link interface) إلى واجهة وصلة مخرج (output link interface) مناسبة. بينما يشير "التوجيه" إلى عملية تتم في كافة أنحاء الشبكة لتحديد المسارات التي تأخذها الرزم من المصدر إلى وجهتها النهائية. وبالتناظر تأمَّل الرحلة من بينسلفانيا إلى فلوريدا التي قام بها المسافر في الجزء 1-3-2. أثناء تلك الرحلة يمر المسافر خلال العديد من المفارق في الطريق حتى يصل إلى فلوريدا. يمكن أن نعتبر "التمرير" كعملية عبور مفرق واحد: تدخل السيارة المفرق من طريق ما ثم تقرّر الطريق الذي يجب أن تأخذه لتغادر المفرق. ويمكن أن نعتبر "التوجيه" كعملية تخطيط الرحلة من بينسلفانيا إلى فلوريدا: فقبل أن يبدأ الرحلة يراجع السائق الخريطة ويختار طريقاً واحداً من بين عدة طرق محتملة. يتكون كل طريق من سلسلة من القطع المتصلة عند المفارق. في هذا الفصل سنفحص أولاً نماذج خدمة طبقة الشبكة، ثم سنركّز على مواضيع طبقة الشبكة المتعلقة بالتمرير، وبعد ذلك نحوّل انتباهنا إلى التوجيه.

يوجد في كل موجّه "جدول تمرير" (forwarding table). عندما تصل رزمة إلى الموجّه يقوم بفحص قيمة حقل في ترويسة الرزمة ويستعملها للبحث في جدول التمرير لديه لتحديد أيٍّ من واجهات الوصلات يجب إرسال الرزمة إليها. ويمكن على حسب بروتوكول طبقة الشبكة المستخدم أن تمثل هذه القيمة في ترويسة الرزمة عنوان وجهة الرزمة أو إشارة تحدد التوصيلة التي تنتمي لها الرزمة. يوضح الشكل 4-2 مثالاً لهذه العملية. ففي هذا الشكل تصل رزمة تحتوي على القيمة 0111 في حقل الترويسة إلى الموجّه. يبحث الموجّه في جدول التمرير لديه، ويقرّر أن واجهة وصلة المخرج لهذه الرزمة هي الواجهة 2، ومن ثمّ يرسل الموجّه تلك الرزمة داخلياً إلى الواجهة 2. في الجزء 4-3 سننظر داخل موجّه ونفحص وظيفة التمرير بتفصيل أكثر.

قد تتساءل الآن كيف تُعَدّ جداول التمرير في الموجّهات؟ إن هذه قضية حاسمة، فهي توضح التفاعل الهام بين التوجيه والتمرير. كما هو مبين في الشكل 4-2 تقرّر خوارزمية التوجيه القيم التي توضع في جداول التمرير بالموجّهات. قد تكون خوارزمية التوجيه مركزية (أي توجد خوارزمية تُنفّذ على موقع مركزي ثم تُحمِّل معلومات التوجيه إلى كل موجّه) أو غير مركزية (أي يوجد جزء من



الشكل 4-2 خوارزميات التوجيه تحدد القيم الموجودة بجداول التمرير.

خوارزمية التوجيه الموزّعة يُنفّذ في كل موجّه). في كلا الحالتين يتلقى الموجّه رسائل بروتوكول التوجيه التي تستعمل لإعداد جدول التمرير لديه. ويمكن أن نوضح الأهداف الأكثر تمييزاً واختلافاً لوظائف التمرير والتوجيه بالنظر في حالة افتراضية (وغير واقعية ولكن ممكنة تقنياً) لشبكة يتم فيها إعداد كل جداول التمرير مباشرة بواسطة "مشغلي الشبكة" الموجودين فعلياً عند الموجّهات. في هذه الحالة لا نحتاج إلى بروتوكولات توجيه! يحتاج مشغلو الشبكة بالطبع للتفاعل مع بعضهم البعض للتأكد من أن جداول التمرير قد أعدت بطريقة تضمن وصول الرزم إلى وجهاتها النهائية. من المحتمل أيضاً أن يكون مثل هذا الإعداد البشري أكثر

عرضة للأخطاء وأبطأ بكثير للاستجابة للتغييرات في طبوغرافية الشبكة من استخدام بروتوكول للتوجيه. ومن حسن الحظ إن كل الشبكات فيها كلا الوظيفتين: التمرير والتوجيه!

وبينما نحن بصدد مصطلحات الشبكات، يجدر بنا التنويه إلى مصطلحين آخرين يستعملان في أغلب الأحيان بشكلٍ متبادل، ولكننا سوف نستعملهما بعناية أكثر. سنخصص المصطلح "محوّل رزم" (packet switch) ليعنى أداة عامة لتحويل الرزم حيث تنقل الرزم من واجهة وصلة المدخل الى واجهة وصلة المخرج طبقاً للقيم الموجودة بحقل ما في ترويسة كل رزمة. تبنى بعض محوّلات الرزم – وتعرف بمحوّلات طبقة ربط البيانات (والتي سنتناولها في الفصل الخامس) – قرار التمرير على القيمة الموجودة في حقل من حقول الترويسة لطبقة ربط البيانات. في حين تبني محوّلات الرزم الأخرى – وتسمى الموجّهات – قرار التمرير على القيمة الموجودة في حقل من حقول ترويسة طبقة الشبكة. (ولتقدير أهمية هذا التمييز قد تحتاج إلى مراجعة الجزء 1-5-2 حيث ناقشنا رزم بيانات طبقة الشبكة وإطارات (frames) طبقة ربط البيانات والعلاقة بينهما). ولأن تركيزنا في هذا الفصل على طبقة الشبكة سنستعمل المصطلح "موجّه" بدلاً من "محوّل الرزم". وحتى عندما نتحدّث عن محوّلات الرزم في شبكات الدائرة الافتراضية (والتي سيتم مناقشتها قريباً) سوف نستعمل المصطلح موجّه.

إعداد التوصيلة (Connection Setup)

ذكرنا للتو أن طبقة الشبكة لها وظيفتان مهمتان: التمرير والتوجيه. ولكننا سنرى قريباً أن في بعض شبكات الحاسب توجد في الحقيقة وظيفة ثالثة مهمة لطبقة الشبكة؛ ألا وهي إعداد التوصيلة. تذكّر من دراستنا لبروتوكول TCP أن المصافحة ثلاثية الاتجاه (three-way handshake) مطلوبة قبل تدفق البيانات من المُرسِل إلى المستقبل. وهذا يسمح للمُرسِل والمُستقبِل بإعداد المعلومات المطلوبة عن الحالة، كالرقم التسلسلي والحجم الابتدائى لنافذة التحكم في التدفق (flow control window). وبأسلوب مماثل تتطلب بعض البنى المعمارية لطبقة الشبكة

(على سبيل المثال شبكة نمط النقل اللاتزامني ATM وشبكة تحويل الإطارات frame relay)، على خلاف شبكة الإنترنت، أن تتصافح الموجّهات على طول المسار المختار من المصدر إلى الوجهة مع بعضها البعض لكي تقوم بإعداد معلومات الحالة قبل أن تبدأ رزم البيانات في التدفق خلال المسار. تسمى هذه العملية في طبقة الشبكة إعداد التوصيلة (وسوف نتناولها في الجزء 4-2).

4-1-2 نماذج الخدمة للشبكة

قبل التنقيب في طبقة الشبكة دعنا نلقِ نظرة أشمل على أنواع الخدمات المختلفة التي قد تقدمها طبقة الشبكة. عندما ترسل طبقة النقل في مضيف الإرسال رزمة إلى الشبكة (أى تمررها الى طبقة الشبكة في مضيف الإرسال) هل يمكن أن تعتمد طبقة النقل على طبقة الشبكة لتسليم الرزمة إلى وجهتها النهائية؟ وعند إرسال رزم متعدّدة هل تصل إلى طبقة النقل في مضيف الاستقبال بنفس ترتيبها عند الإرسال؟ وهل مقدار الوقت بين إرسال رزمتين متتاليتين هو نفسه تماماً مقدار الوقت بين استقبالهما؟ وهل توفر الشبكة أي تغذية مرتجعة (feedback) حول الازدحام (congestion) في الشبكة؟ وما هو التمثيل المجرّد (أي الصفات) للقناة التي تربط بين طبقة النقل في مضيف الإرسال ومضيف الاستقبال؟ تتحدد الإجابة على هذه الأسئلة وأسئلة أخرى بنموذج الخدمة الذي توفره طبقة الشبكة. يُعرِّف نموذج خدمة الشبكة خصائص نقل الرزم من طرف إلى طرف بين حافة وأخرى للشبكة (أي بين الأنظمة الطرفية للإرسال والاستقبال).

دعنا الآن نراجع بعض الخدمات المحتملة التي يمكن أن توفرها طبقة الشبكة. عندما ترسل طبقة النقل رزمة إلى طبقة الشبكة في مضيف الإرسال فإن الخدمات التي يمكن أن توفرها طبقة الشبكة تشمل:

- ضمان التوصيل: تضمن هذه الخدمة وصول الرزمة في النهاية إلى وجهتها.
- ضمان التوصيل مع تأخير محدود: هذه الخدمة لا تضمن فقط توصيل الرزمة ولكن توصيلها في زمن تأخير محدد من مضيف إلى مضيف (مثلاً في خلال 100 ميلى ثانية).

كما يمكن أن توفر طبقة الشبكة علاوةً على ذلك الخدمات التالية لتدفق الرزم بين مصدر معين ووجهة معينة:

- توصيل الرزم بنفس الترتيب: تضمن هذه الخدمة وصول الرزم إلى وجهتها بنفس الترتيب الذي أرسلت به.
- ضمان الحد الأدنى للحيز الترددي: تحاكي هذه الخدمة في طبقة الشبكة سلوك وصلة إرسال ذات معدل بيانات محدد (على سبيل المثال ميجابت واحد في الثانية) بين مضيفات الإرسال والاستقبال (بالرغم من أن المسار الفعلي من طرف إلى طرف قد يتكون من عدّة وصلات مادية). طالما يرسل المضيف البتات (كجزء من الرزم) بمعدل أقل من المعدل المحدّد فإن الرزم لا تفقد، وتصل كل رزمة في غضون تأخير "من مضيف إلى مضيف" محدد مسبقاً (مثلاً خلال 40 ميلي ثانية).
- ضمان الحد الأقصى للتفاوت الزمني للتأخير (delay jitter): تضمن هذه الخدمة أن يكون مقدار الوقت بين رزمتين متتاليتين عند المُرسِل يساوي مقدار الوقت بينهما عند الوجهة النهائية (أو ألا تتجاوز التغيرات في الفترة الزمنية بينهما قيمة محدّدة).
- خدمات الأمن: باستعمال مفتاح جلسة سري معروف فقط للمصدر والوجهة يمكن لطبقة الشبكة في مضيف المصدر أن تشفّر (encrypt) بيانات كل رزمة تُرسلها إلى مضيف الوجهة. وستكون طبقة الشبكة في مضيّف الوجهة مسؤولة عن حل الشفرة (decrypt) واسترجاع الشكل الأصلي للبيانات. بمثل هذا الخدمة ستُضمن السريّة (الخصوصية) لكل قطع بيانات طبقة النقل (TCP وUDP) بين مضيفي المصدر والوجهة. وبالإضافة إلى السريّة يمكن أن توفر طبقة الشبكة خدمات أخرى كسلامة البيانات (data integrity) والتوثيق (authentication) للتحقق من المصدر.

هذه فقط قائمة جزئية من الخدمات التي يمكن أن توفرها طبقة الشبكة؛ فهناك العديد من الاختلافات الممكنة التي لا تحصى.

توفر طبقة الشبكة في الإنترنت خدمة واحدة تعرف بخدمة "أفضل جهد" (best-effort). يتضح من الجدول 4-1 أن خدمة "أفضل جهد" هي تعبير تلطيفي لـ "لا خدمة على الاطلاق". فمع هذه الخدمة لا ضمان للإبقاء على الوقت بين الرزم، ولا ضمان لتوصيل الرزم المرسلة بنفس الترتيب، ولا ضمان لتوصيلها نهائياً. بهذا التعريف فإن الشبكة التي لم توصّل أي رزمة إلى الوجهة توافق تعريف خدمة أفضل جهد للتوصيل. ومع ذلك فكما سنناقش بعد قليل هناك أسباب معقولة وراء وجود هذا النموذج لخدمة الحد الأدنى لطبقة الشبكة. وسوف نغطّي نماذج خدمة إضافية – ما زالت في مرحلة التطوير – للإنترنت في الفصل السابع.

البنية المعمارية للشبكة	نموذج الخدمة	الحيز الترددي	عدم الفقد	الترتيب	التوقيت	الإشارة إلى الازدحام
الإنترنت	أفضل جهد	غير مضمون	غير مضمون	أي ترتيب	غير مضمون	غير متوفر
ATM	معدل ثابت للبتات (CBR)	المعدل الثابت مضمون	مضمون	مضمون	مضمون	لا يحدث ازدحام
ATM	معدل البتات المتاح (ABR)	المعدل الأدنى مضمون	غير مضمون	مضمون	غير مضمون	متوفر

الجدول 4-1 نماذج الخدمة في الإنترنت وشبكة ATM.

هناك أيضاً بنيات معمارية أخرى لشبكات تُعرِّف وتُحقِّق نماذج خدمة تتجاوز خدمة أفضل جهد في الإنترنت، لكنها خارج نطاق هذا الكتاب. على سبيل المثال توفر بنية شبكة ATM [ATM Forum 2007; Black 1995] عدة نماذج للخدمة وهذا يعني أنه يمكن عمل اتصالات بأنواع مختلفة للخدمة ضمن نفس الشبكة. إن مناقشة كيفية توفير شبكة ATM لمثل هذه الخدمات تقع خارج نطاق هذا الكتاب، فهدفنا هنا فقط هو التنويه عن وجود بدائل لنموذج "خدمة أفضل جهد" المستخدم في الإنترنت. اثنان من نماذج خدمة ATM الأكثر أهميّة هما: خدمة معدل البتات الثابت (Constant Bit Rate (CBR)) وخدمة معدل البتات المتوفر (Available Bit Rate (ABR)):

- خدمة معدل البتات الثابت (CBR) لشبكة ATM: كان هذا أول نموذج خدمة قياسي لشبكة ATM، وهو يعكس اهتماماً مبكّراً من شركات الهاتف بشبكة ATM ومدى ملاءمة خدمة CBR لنقل بيانات الصوت والفيديو ذات المعدل الثابت في الوقت الحقيقي. إن هدف خدمة CBR بسيط من حيث المفهوم: ألا وهو توفير تدفق من الرزم (المعروفة بالخلايا (cells) في مصطلحات ATM) خلال أنبوب افتراضي له نفس الخواص تماماً كما لو كان وصلة إرسال مخصصة ذات حيز ترددي ثابت بين المضيفين المُرسِل والمستقبل. ومع خدمة CBR يُحمَل تدفق من خلايا ATM عبر الشبكة بطريقة بحيث يضمن أن يكون تأخير الخلايا من طرف إلى طرف (end-to-end delay) والتفاوت الزمني للتأخير (delay jitter) ومعدل فقد الخلايا أقل من قيم محدّدة. هذه القيم يُتّفق عليها بين المضيف المُرسِل وشبكة ATM عند بداية إرساء اتصال بمعدل إرسال ثابت (CBR).
- خدمة معدل البتات المتوفر (ABR) لشبكة ATM: في حين تقدم الإنترنت ما يسمى بخدمة أفضل جهد، فإن خدمة ABR لشبكة ATM يمكن تمييزها على أنها تعديل بعض الشيء لخدمة أفضل جهد. وكما هو الحال مع نموذج خدمة الإنترنت قد تُفقد الخلايا مع خدمة ABR. لكن على خلاف الإنترنت فإنه لا يمكن أن يختلف ترتيب الخلايا (بالرغم من أن البعض قد يفقد)، والمعدل الأدنى لإرسال الخلايا (Minimum Cell Rate (MCR)) مضمون لاتصال يستخدم خدمة ABR. وإذا كانت الموارد المتاحة في الشبكة في وقت ما كافية يمكن للمُرسِل أيضاً إرسال الخلايا بنجاح بنسبة أعلى من MCR. إضافة إلى ذلك — كما رأينا في الجزء 3-6 – يمكن أن توفر خدمة ABR في شبكة ATM تغذية مرتجعة إلى المُرسِل على شكل بت إخطار الازدحام (congestion notification bit) أو معدل إرسال صريح (explicit transmission rate) لضبط معدل الإرسال بين MCR والمعدل الأقصى المسموح به.

2-4 شبكات الدائرة الافتراضية وشبكات وحدات البيانات

تذكّر من الفصل الثالث أن طبقة النقل يمكن أن تقدم للتطبيقات خدمة لاتوصيلية (connectionless) أو خدمة توصيلية (connection-oriented). على سبيل المثال تزوّد طبقة النقل في الإنترنت كل تطبيق بإمكانية الاختيار بين خدمتين: UDP (خدمة لاتوصيلية) أو TCP (خدمة توصيلية). وبطريقة مماثلة يمكن أن تقدّم طبقة الشبكة أيضاً خدمة لاتوصيلية أو خدمة توصيلية، وهي توازي خدمات طبقة النقل من عدّة أوجه. على سبيل المثال تبدأ الخدمة التوصيلية في طبقة الشبكة بالمصافحة (handshaking) بين مضيفي المصدر والوجهة، بينما لا توجد أية تمهيدات للمصافحة في خدمة طبقة الشبكة اللاتوصيلية.

بالرغم من وجود بعض أوجه التشابه بين خدمات طبقة الشبكة اللاتوصيلية والتوصيلية مع خدمات طبقة النقل إلا أن هناك اختلافات جوهرية:

- تكون الخدمات في طبقة الشبكة من مضيف إلى مضيف وتوفرها طبقة الشبكة لطبقة النقل. أما في طبقة النقل فتكون الخدمات من عملية إلى عملية وتوفرها طبقة النقل لطبقة التطبيقات.
- في كل البنى الرئيسة لشبكات الحاسب حتى الآن (الإنترنت، وشبكة ATM، وشبكة تحويل الإطارات، وغيرها) توفر طبقة الشبكة إمّا خدمة لاتوصيلية من مضيف إلى مضيف أو خدمة توصيلية من مضيف إلى مضيف؛ لكن ليسا معاً. تسمى شبكات الحاسب التي تقدّم خدمة توصيلية فقط في طبقة الشبكة شبكات الدائرة الافتراضية ((Virtual Circuit (VC)؛ في حين تسمى شبكات الحاسب التي تقدّم خدمة لا توصيلية فقط في طبقة الشبكة شبكات وحدات البيانات (datagram networks).
- يختلف تحقيق الخدمة التوصيلية في طبقة النقل بشكلٍ أساسي عن تحقيقها في طبقة الشبكة. سبق أن رأينا في الفصل السابق أن خدمة طبقة النقل التوصيلية مطبّقة في حافة الشبكة في الأنظمة الطرفية، لكن كما سنرى بعد قليل توجد الخدمة التوصيلية في طبقة الشبكة في الموجهّات الموجودة في قلب الشبكة بالإضافة إلى وجودها في الأنظمة الطرفية.

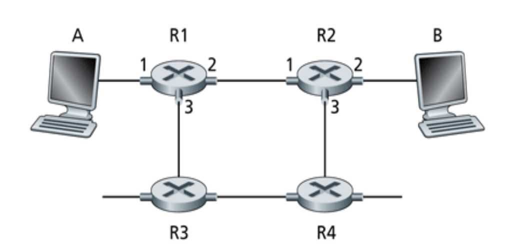
إن شبكات الدائرة الافتراضية وشبكات وحدات البيانات نوعان أساسيان لشبكات الحاسب، وهما يستعملان معلومات مختلفة جداً في اتخاذ قرارات التمرير. دعنا الآن نلقِ نظرة أقرب في كيفية تحقيقهما.

4-2-1 شبكة الدائرة الافتراضية

لقد عرفنا أن الإنترنت هي إحدى شبكات وحدات البيانات (datagram networks). ولكن هناك العديد من بنى الشبكات المعمارية البديلة – بما في ذلك شبكة ATM وشبكة تحويل الإطارات – تستخدم الدائرة الافتراضية، ولذا فهي تستعمل التوصيلات في طبقة الشبكة، والتي يطلق عليها "دوائر افتراضية" (VCs). دعنا الآن نرى كيفية تحقيق خدمة الدائرة الافتراضية VC في شبكات الحاسب.

تتكون الدائرة الافتراضية من: (1) مسار (أي سلسلة من الوصلات والموجّهات) بين مضيفي المصدر والوجهة، (2) أرقام الدائرة الافتراضية وتشمل رقماً واحداً لكل وصلة على طول المسار، (3) مدخلات في جدول التمرير في كل موجّه على طول المسار. تحمل كل رزمة تنتمي لدائرة افتراضية رقماً للدائرة الافتراضية (VC number) في ترويستها. ولأن الدائرة الافتراضية قد يكون لها رقم مختلف على كل وصلة فعند مرور الرزمة على المسار يجب أن يستبدل كل موجّه بيني رقم الدائرة الافتراضية برقم جديد يحصل عليه من جدول التمرير.

لتوضيح هذا المفهوم انظر إلى الشبكة الموضحة في الشكل 4-3. تمثل الأرقام الموجودة بجانب الوصلات للموجّه R1 أرقام واجهات الوصلات للموجّه. افترض الآن أن المضيف A يطلب من الشبكة إعداد دائرة افتراضية VC بينه وبين المضيف B. وافترض أيضاً أن الشبكة تختار المسار A-R1-R2-B وتخصص الأرقام 12، 22، 32 للوصلات الثلاث في هذا المسار لهذه الدائرة الافتراضية. في هذه الحالة عندما تغادر رزمة المضيف A تكون القيمة في حقل رقم الدائرة الافتراضية في ترويسة الرزمة 12، وعندما تغادر R1 تصبح القيمة 22، وعندما تغادر R2 تصبح القيمة 32.



الشكل 4-3 مثال بسيط لشبكة الدائرة الافتراضية.

كيف يُقرّر الموجّه رقم الدائرة الافتراضية الجديد لرزمة تعبر خلاله؟ في شبكة الدائرة الافتراضية يتضمّن كل جدول تمرير للموجّه ترجمة لأرقام الدوائر الافتراضية. على سبيل المثال جدول التمرير في الموجّه R1 قد يبدو مثلاً كما يلي:

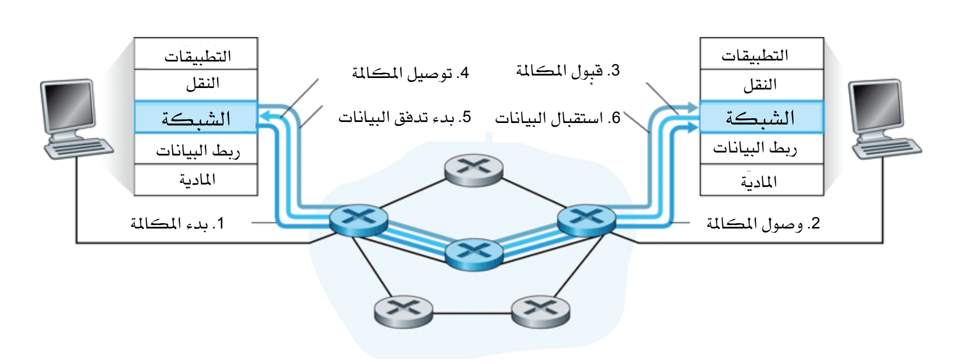
واجهة المدخل	رقم الدائرة الافتراضية للمدخل	واجهة المخرج	رقم الدائرة الافتراضية للمخرج
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

عندما تؤسّس VC جديدة عبر موجّه يضاف مُدخل جديد إلى جدول التمرير لديه. وبنفس الطريقة عند إنهاء VC تُزَال المُدخلات المتعلقة بها من كل جداول التمرير على طول مسارها.

قد تتساءل لماذا لا تحتفظ الرزمة بنفس رقم VC على كل وصلة من الوصلات على طول المسار. يرجع ذلك إلى سببين. الأول هو أن تغيير الرقم من وصلة إلى وصلة يقلل طول حقل VC في ترويسة الرزمة. والثاني – وهو الأهم – أن عملية إعداد VC تكون أبسط بكثير عند السماح لرقم VC بالتغيّر لكل وصلة على طول مسار VC. وبالتحديد باستعمال أرقام VC متعدّدة يمكن أن تختار كل وصلة على المسار رقم VC بشكلٍ مستقل عن أرقام VC التي يتم اختيارها على الوصلات الأخرى على طول المسار. أما إذا تطلبنا أن يكون رقم VC ثابتاً لكل الوصلات على طول

المسار فإن الموجهات يجب أن تتبادل وتعالج عدداً كبيراً من الرسائل للموافقة على رقم مشترك (مثلاً رقم يكون غير مستخدم من قِبَل أي دائرة افتراضية أخرى موجودة حالياً في تلك الموجّهات) لكي يُستعمَل لهذه التوصيلة الجديدة.

في شبكة VC يجب أن تحتفظ موجّهات الشبكة بمعلومات حالة عن التوصيلات الموجودة حالياً. بالتحديد في كل مرة يتم تأسيس توصيلة جديدة عبر موجّه يجب أن يضاف مُدخل جديد عن التوصيلة إلى جدول التمرير، وفي كل مرة يتم إنهاء توصيلة يجب أن يحذف المُدخل المتعلق بها من الجدول. لاحظ أنه حتى في حالة عدم وجود ترجمة لأرقام الدوائر الافتراضية ما زال من الضروري الاحتفاظ بمعلومات حالة عن التوصيلات تقرن أرقام VC بأرقام واجهات المخرج. تُعتبر قضية احتفاظ الموجّه أو عدم احتفاظه بمعلومات حالة لكل توصيلة موجودة حالياً من القضايا الهامة والتي سنعود إليها مراراً وتكراراً في هذا الكتاب.



الشكل 4-4 إعداد دائرة افتراضية.

هناك ثلاث مراحل مميّزة للدائرة الافتراضية:

- إعداد VC: أثناء مرحلة إعداد VC تتصل طبقة النقل للمُرسِل بطبقة الشبكة، وتحدد عنوان المستقبل، وتنتظر حتى تقوم الشبكة بإعداد VC. تحدد طبقة الشبكة المسار بين المُرسِل والمستقبل، أي سلسلة الوصلات والموجّهات التي تمر خلالها كل رزم الدائرة الافتراضية VC. كما تحدد طبقة الشبكة أيضاً رقم VC لكل وصلة على طول المسار. وفي النهاية

تضيف طبقة الشبكة مُدخلاً في جدول التمرير في كل موجّه على طول المسار. أثناء إعداد VC قد تحجز طبقة الشبكة أيضاً الموارد اللازمة (على سبيل المثال الحيز الترددي) على طول مسار الدائرة الافتراضية.

- نقل البيانات: كما هو موضح في الشكل 4-4 بمجرد إعداد دائرة افتراضية يمكن أن تبدأ الرزم بالتدفق خلال تلك الدائرة الافتراضية.
- إنهاء (فض) الدائرة الافتراضية: تبدأ هذه الخطوة عندما يخبر المُرسِل (أو المستقبل) طبقة الشبكة عن رغبته في إنهاء الدائرة الافتراضية. بعد ذلك تخبر طبقة الشبكة عادةً النظام الطرفي على الجانب الآخر للشبكة لإنهاء الاتصال وتُعدَّل جداول التمرير في كل الموجّهات على المسار للإشارة إلى أن الدائرة الافتراضية لم تعد قائمة.

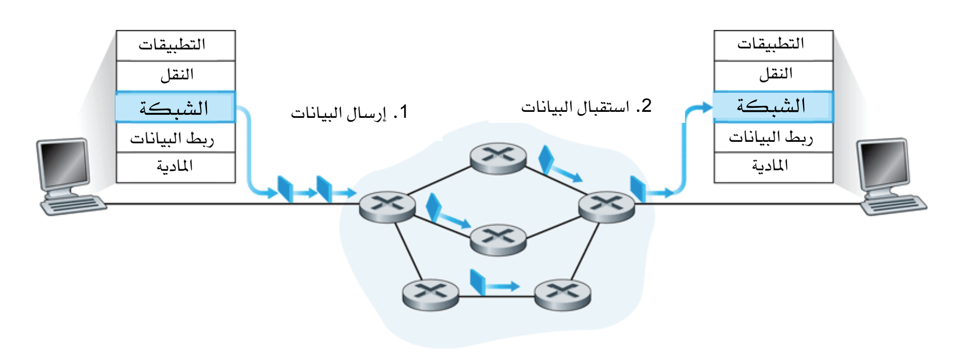
هناك اختلاف دقيق ولكنه مهم بين إعداد VC في طبقة الشبكة وإعداد التوصيلة في طبقة النقل (على سبيل المثال المصافحة الثلاثية لبروتوكول TCP الذي درسناه في الفصل الثالث). يقتصر إعداد التوصيلة في طبقة النقل على النظامين الطرفيين فقط، فهما وحدهما يحددان البارامترات المطلوبة لتوصيلة طبقة النقل (على سبيل المثال الرقم التسلسلي الأولي وحجم نافذة التحكم في التدفق). ورغم أن النظامين الطرفيين يكونان على دراية بتوصيلة طبقة النقل، إلا أن الموجّهات على طول المسار بين النظامين الطرفيين تكون غافلة عنها تماماً. في المقابل في شبكة الدوائر الافتراضية تشترك كل الموجّهات على طول المسار بين النظامين الطرفيين في إعداد الدائرة الافتراضية، ويكون كل موجّه على دراية تامة بكل الدوائر الافتراضية التي تعبره.

وتُعرَف الرسائل التي ترسلها الأنظمة الطرفية إلى الشبكة لبدء أو إنهاء VC، والرسائل التي تعبر بين الموجّهات لبدء VC (أي لتَعديل حالة الاتصال في جداول الموجّه) باسم "رسائل التحكم" (رسائل التأشير) (signaling messages)، وغالباً ما تسمى البروتوكولات المستخدمة لتبادل تلك الرسائل بروتوكولات التحكم (بروتوكولات التأشير). يوضح الشكل 4-4 إعداد VC بشكلٍ تصويري. سوف لا نغطي بروتوكولات التأشير للدوائر الافتراضية في هذا الكتاب. راجع [Black

1997] لمناقشة عامة عن التأشير في الشبكات التوصيلية، وراجع معيار الاتحاد الدولي للمواصلات السلكية واللاسلكية [ITU-T Q.2931 1994] لمواصفات بروتوكول التأشير Q.2931 المستخدم في شبكات ATM.

4-2-2 شبكات وحدات البيانات (Datagram Networks)

في كل مرة يريد نظام طرفي في شبكة وحدات البيانات إرسال رزمة يختم الرزمة بعنوانِ النظام الطرفي للوجهة ثم يدفعها إلى الشبكة. يتم ذلك بدون أي إعداد للدوائر الافتراضية كما هو موضح في الشكل 4-5. ولا تحتفظ الموجّهات في شبكة وحدات البيانات بأية معلومات حالة حول الدوائر الافتراضية (لأنها لا توجد أصلاً!).



الشكل 4-5 شبكة وحدات البيانات.

وبينما تنتقل رزمة من المصدر إلى الوجهة فإنها تمر خلال سلسلة من الموجّهات. تَستعمل كلٌّ من تلك الموجّهات عنوان الوجهة في ترويسة الرزمة لتمريرها. وبشكلٍ مُحدّد، كل موجّه له "جدول تمرير" (forwarding table) يترجم عناوين الوجهة النهائية إلى واجهات الوصلات عليه. وعندما تصل رزمة إلى الموجّه، يستعمل الموجّه عنوان الوجهة النهائية لها للبحث عن واجهة وصلة المخرج الملائمة في جدول التمرير لديه، ثم يرسلها الموجّه عمداً إلى تلك الواجهة.

لفهم عملية البحث في الجدول (lookup) بشكلٍ أفضل، دعنا ننظر إلى مثالٍ محدد. افترض أن عناوين كل الوجهات تتكون من 32 بتاً (وهو نفس طول عنوان الوجهة في رزمة بيانات IP). تقتضي الطريقة المباشرة لتكوين جدول التمرير وجود مُدخل واحد في الجدول لكل عنوان محتمل للوجهة. ولأن هناك أكثر من 4 بلايين عنوان محتمل فهذا الخيار غير ممكن عملياً لأنه يتطلب جدول تمرير ضخم للغاية.

الآن دعنا نفترض بعد ذلك أن موجّهنا له أربع وصلات مرقمة من 0 إلى 3، وأن الرزم سترسل إلى واجهات الوصلات كالتالي:

واجهة الوصلة	مدى عناوين الوجهة
0	11001000 00010111 00010000 00000000 حتى 11001000 00010111 00010111 11111111
1	11001000 00010111 00010000 00000000 حتى 11001000 00010111 00011000 11111111
2	11001000 00010111 00011001 00000000 حتى 11001000 00010111 00011111 11111111
3	ما عدا ذلك

ومن الواضح في هذا المثال أنه ليس من الضروري وجود 4 بلايين مُدخل في جدول التمرير للموجّه. يمكن أن نستخدم على سبيل المثال جدول التمرير التالي بأربعة مُدخلات فقط:

واجهة الوصلة	تطابق البادئة
0	11001000 00010111 00010
1	11001000 00010111 00011000
2	11001000 00010111 00011
3	ما عدا ذلك

بهذا الأسلوب لجدول التمرير يطابق الموجّه بادئة (prefix) في عنوان الوجهة للرزمة بالمُدخلات في الجدول. إذا حدث تطابق بين عنوان الوجهة للرزمة وأحد مُدخلات الجدول يرسل الموجّه الرزمة إلى الوصلة المقترنة بذلك التطابق. على سبيل المثال افترض أن عنوان وجهة الرزمة هو 10100001 00010110 00010111 11001000. لأن بادئة العنوان المكونة من 21 بتاً توافق أول مُدخل في الجدول سوف يرسل الموجّه الرزمة لواجهة الوصلة رقم 0. أما إذا لم يوجد تطابق مع أيٍّ من المُدخلات الأولى الثلاثة فإن الموجّه سيُرسِل الرزمة إلى الواجهة رقم 3. وبالرغم من أن هذه الطريقة تبدو بسيطة للغاية إلا أنه يوجد هنا معنى دقيق ومهم. ربما لاحظت أنه من المحتمل أن يطابق عنوان الوجهة أكثر من مُدخل واحد في الجدول. على سبيل المثال الـ 24 بتاً الأولى من العنوان 10101010 00011000 00010111 11001000 تطابق المُدخل الثاني في الجدول، والـ 21 بتاً الأولى تطابق المُدخل الثالث في الجدول. عند وجود تطابقات متعدّدة يستعمل الموجّه قاعدة تطابق البادئة الأطول (longest prefix matching rule)، وهذا يعني تحديد المُدخل الذي يحقق أطول تطابق في الجدول ويرسل الرزمة إلى واجهة الوصلة المقترنة به.

بالطبع لكي تكون هذه القاعدة فعّالة يجب أن تكون كل واجهة وصلة مخرج مسؤولة عن تمرير كتل كبيرة من عناوين متجاورة للوجهات. سنرى في الجزء 4-4 أن عناوين الإنترنت عادة ما تخصّص بطريقة هرمية لكي تكون خاصية "التجاور" هذه سائدة في جداول التمرير لأكثر الموجّهات. وعلى الرغم من هذا هناك بعض القلق في المجتمع البحثي للإنترنت حول وجود ثقوب (عناوين غير مستخدمة) أكثر في فضاء العناوين يسبّب صغر الكتل المتجاورة أكثر، وبالتالى تصبح جداول التمرير أكبر (راجع [Meng 2005]، و[RFC 3221]، والمناقشة "المبادئ في الواقع العملي" في الجزء 4-4).

رغم أن الموجّهات في شبكات وحدات البيانات لا تحتفظ بأية معلومات عن حالة التوصيلة إلا أنها تحتفظ بمعلومات عن حالة التمرير في جداول التمرير. لكن المعدّل الزمني الذي تتغيّر فيه معلومات الحالة هذه بطيء نسبياً. في الواقع يتم تعديل جداول التمرير في شبكة وحدات البيانات بواسطة خوارزميات التوجيه، وعادة ما

يتم ذلك على فترات تتراوح من دقيقة إلى خمس دقائق أو نحوها. أما في شبكة الدائرة الافتراضية فيتم تعديل الجدول في الموجّه عند بدء توصيلة جديدة خلاله أو انتهاء توصيلة موجودة حالياً خلاله. وهذا يمكن أن يحدث بسهولة في مقياس زمني بالميكروثانية (جزء من مليون من الثانية) في موجّهات المستوى الأول لشبكة العمود الفقري للإنترنت (backbone tier-1 router).

ولأنه يمكن أن تُعدَّل جداول التمرير في شبكات وحدات البيانات في أي وقت فإن سلسلة من الرزم المرسلة من نظام طرفي إلى آخر قد تتبع مسارات مختلفة خلال الشبكة وقد تصل بترتيب مختلف. وقد قدَّم [Paxson 1997] و[Jaiswal 2003] دراسات لقياس إعادة ترتيب الرزم وظواهر أخرى في الإنترنت العامة.

4-2-3 نشأة شبكات الدوائر الافتراضية وشبكات وحدات البيانات

يعكس تطور شبكات وحدات البيانات وشبكات VC منشأها. ففكرة الدائرة الافتراضية كمبدأ تنظيم مركزي لها جذورها في عالم اتصالات الهاتف الذي يستعمل الدوائر الحقيقية. فبإعداد التوصيلة، والاحتفاظ بمعلومات عن حالة كل توصيلة في الموجّهات الموجودة في الشبكة، تصبح شبكة VC جدلياً أكثر تعقيداً من شبكة وحدات البيانات؛ وهذا يتوافق أيضاً مع جذورها المتمثلة في شبكات الاتصالات الهاتفية. ويمكنك الاطلاع على [Molinero-Fernandez 2002] لمقارنة هامة بين مدى تعقيد شبكات تحويل الدوائر (circuit-switched networks) وشبكات تحويل رزم البيانات (packet-switched networks). فقد كان التعقيد في شبكة الهواتف بالضرورة ضمن الشبكة، لأنها كانت توصّل بين أجهزة أنظمة طرفية غير ذكية كالهواتف الدوّارة (rotary telephones) — ولأولئك الشباب الذين لا يعرفون الهاتف الدوّار هو هاتف تناظري (analog telephone) بدون أزرار؛ فقط يوجد قرص دوَّار (dial).

ومن ناحية أخرى نشأت الإنترنت (كشبكة وحدات بيانات) لسد الحاجة لتوصيل الحاسبات مع بعضها. ومع أجهزة أنظمة طرفية أكثر تطوراً اختار مصمّمو الإنترنت أن يجعلوا نموذج خدمة طبقة الشبكة بسيطاً قدر الإمكان. كما رأينا في

الفصلين الثانى والثالث تم تطبيق وظائف إضافية (على سبيل المثال توصيل الرزم بنفس الترتيب، والنقل الموثوق للبيانات ، والتحكم في الازدحام، وترجمة أسماء النطاقات DNS) في طبقة أعلى في الأنظمة الطرفية. وقد كان لهذا النموذج عدة نتائج مثيرة:

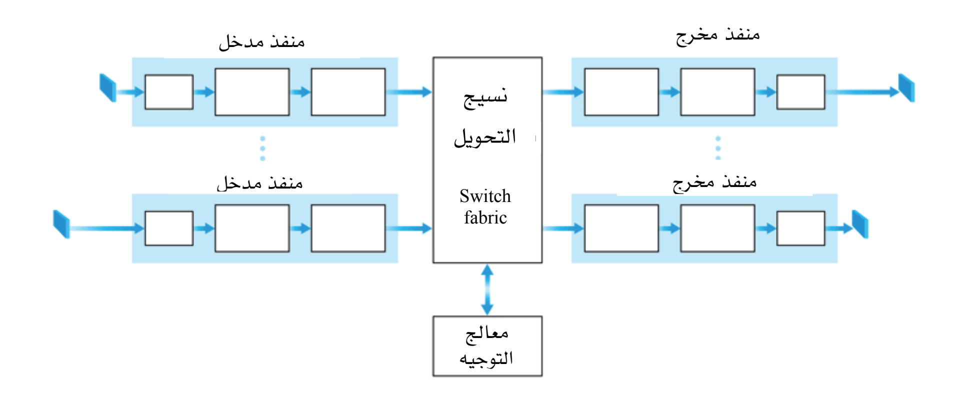
- يُسهِّل نموذج خدمة طبقة شبكة الإنترنت الناتج والذي يوفر أقل ضمانات للخدمة (أو لا ضمانات في الواقع!) (ومن ثم يفرض الحد الأدنى من المتطلبات على طبقة الشبكة) ربط الشبكات التي تستعمل تقنيات مختلفة جداً لطبقة ربط البيانات (على سبيل المثال الأقمار الصناعية، والإيثرنت، والألياف الضوئية، وموجات الراديو) ولها خصائص معدلات إرسال ونسب فقد مختلفة جداً. سوف ندرس توصيل شبكات IP بالتفصيل في الجزء 4-4.
- كما رأينا في الفصل الثانى يتم تحقيق التطبيقات كالبريد الإلكتروني والويب وحتى خدمة مركزية لطبقة الشبكة مثل DNS في المضيفات (الخادمات) على حافة الشبكة. وقد أدت تلك القدرة على إضافة خدمة جديدة عن طريق ربط المضيف بالشبكة وتعريف بروتوكول جديد لطبقة الشبكة (مثل HTTP) إلى انتشار استعمال التطبيقات الجديدة كالويب على الإنترنت في فترة زمنية قصيرة جداً.

كما سنرى في الفصل السابع يسود جدلٌ حادٌّ في مجتمع الإنترنت حول كيفية تطوّر البنية المعمارية لطبقة الشبكة في الإنترنت لكي تدعم الخدمات الفورية (real time) كتطبيقات الصوت والصورة. وتوجد مقارنة هامة بين البنية المعمارية لشبكة ATM المعتمدة على الدوائر الافتراضية ومقترح للبنية المعمارية للجيل القادم للإنترنت في [Crowcroft 1995].

4-3 ماذا بداخل الموجّه؟

الآن وبعد أن رأينا مخططاً عاماً لوظائف وخدمات طبقة الشبكة دعنا نحوّل انتباهنا إلى وظيفة "التمرير" لطبقة الشبكة (أي النقل الفعلي للرزم داخل الموجّه من وصلات المدخل إلى وصلات المخرج المناسبة). ولقد ألقينا نظرة سريعة حول بضع

قضايا لوظيفة التمرير في الجزء 4-2، وبالتحديد العنونة وتطابق أطول بادئة. سنتناول في هذا الجزء البنية المعمارية لموجّه معيّن لنقل الرزم من وصلات المدخل إلى وصلات المخرج. وقد تعمدنا الاختصار لأن تغطية تصميم موجّه بتعمّق يحتاج إلى منهج كامل. وبالتالي سنبذل جهداً خاصاً في هذا الجزء لتزويد القارئ ببعض المراجع التي تغطّي هذا الموضوع بتعمق أكثر. ونُذكّر هنا بأن المصطلحين "تمرير" (forwarding) و"تحويل" (switching) يستعملان في أغلب الأحيان بالتبادل من قِبَل الباحثين والعاملين في حقل شبكات الحاسب، وسوف نستعملهما في هذا الكتاب الدراسي.



الشكل 4-6 البنية المعمارية للموجّه.

يبين الشكل 4-6 مخططاً عالي المستوى لبنية معمارية عامة للموجّه؛ ومنه يمكن تمييز أربعة مكوّنات رئيسة للموجّه:

- منافذ المدخل: تؤدّي هذه المنافذ عدّة وظائف. فهي تقوم بوظائف الطبقة المادية (الصندوق الموجود في أقصى اليسار لمنافذ المدخل، وأقصى اليمين لمنافذ المخرج في الشكل 4-6) عند نهاية وصلة مادية قادمة للموجّه، وتؤدّي وظائف طبقة ربط البيانات (ممثّلة بالصناديق الموجودة في المنتصف لمنافذ المدخل والمخرج) والمطلوبة للتعامل مع وظائف طبقة ربط البيانات في الجانب البعيد للوصلة القادمة، كما تؤدّي أيضاً وظائف البحث في الجدول والتمرير

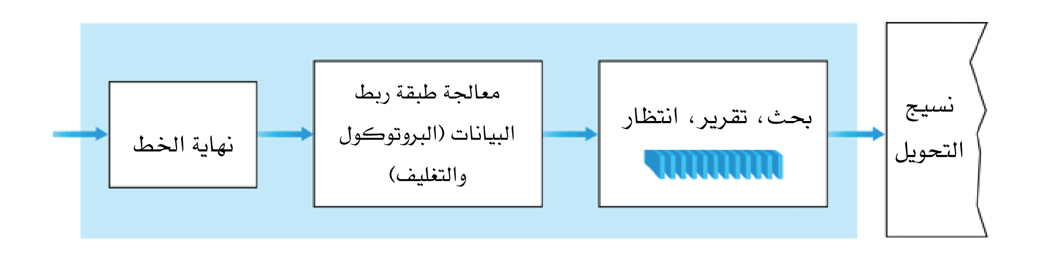
(الصندوق الموجود في أقصى اليمين لمنافذ المدخل وأقصى اليسار لمنافذ المخرج) لكي تخرج كل رزمة مرسلة إلى نسيج التحّويل للموجّه من منفذ المخرج المناسب لها. وترسل رزم التحكم (على سبيل المثال الرزم التي تحمل معلومات بروتوكول التوجيه) من منفذ المدخل إلى معالج التوجيه (routing processor). وعملياً تتجمع عدة منافذ في أغلب الأحيان على "بطاقة خط" (line card) واحدة داخل الموجّه.

- نسيج التحويل: يوصّل نسيج التحويل منافذ المدخل للموجّه بمنافذ المخرج. يوجد نسيج التحويل بالكامل داخل الموجّه (شبكة داخل الموجّه!).
- منافذ المخرج: يخزّن منفذ المخرج الرزم التي أرسلت إليه خلال نسيج التحويل، ثم بعد ذلك يرسلها على وصلة المخرج. وهكذا يؤدّي منفذ المخرج الوظائف العكسية لطبقة ربط البيانات والطبقة المادية لمنفذ المدخل. وعندما تكون الوصلة مزدوجة الاتجاه (أي تحمل بيانات في كلا الاتجاهين) فإن منفذ المخرج إلى الوصلة عادة ما يتزاوج مع منفذ المدخل لتلك الوصلة على نفس بطاقة الخط.
- معالج التوجيه: ينفّذ معالج التوجيه بروتوكولات التوجيه (على سبيل المثال البروتوكولات التي سندرسها في الجزء 4-6)، ويحتفظ بمعلومات التوجيه وجداول التمرير، ويؤدّي وظائف إدارة الشبكة (انظر الفصل التاسع) داخل الموجّه.

في الأجزاء التالية سوف ننظر بتفصيل أكثر إلى منافذ المدخل، ونسيج التحويل، ومنافذ المخرج. راجع [Chuang 2005; Keslassy 2003; Chao 2001; Turner 1988; Giacopelli 1990; McKeown 1997a; Partridge 1998] لمناقشة لبعض البنى المعمارية المحددة للموجّهات. قدّم [McKeown 1997b] نظرة عامة سهلة القراءة للبنى الحديثة للموجّهات مستخدماً موجّه Cisco 1200 كمثال. وللدقة تفترض المناقشة التالية أن شبكة الحاسب شبكة رزم، وأن قرارات التمرير مستندة على عنوان وجهة الرزمة (بدلاً من رقم VC في شبكة الدائرة الافتراضية). ومع ذلك فالمفاهيم والأساليب مماثلة لشبكة الدائرة الافتراضية.

4-3-1 منافذ المدخل

يبين الشكل 4-7 رؤية أكثر تفصيلاً لوظائف منفذ المدخل. كما ذكرنا آنفاً فإن وظائف منفذ المدخل في توفير النهاية للخط ومعالجة وصلة البيانات هي تحقيق للطبقة المادية وطبقة ربط البيانات الخاصة بذلك المنفذ للموجّه. وتُعتبر وحدة البحث في الجدول والتمرير الموجودة في منفذ المدخل أساسية لوظيفة التمرير في الموجّه. في العديد من الموجّهات، يتم هنا تحديد منفذ المخرج الذي سترسل إليه الرزمة الواصلة وذلك عن طريق نسيج التحويل. ويتم اختيار منفذ المخرج باستعمال المعلومات الموجودة في جدول التمرير. ورغم أن جدول التمرير يُحسب بمعالج التوجيه، تُخزن نسخة ظلّ (shadow copy) عادةً من جدول التمرير في كل منفذ مدخل وتُحدَّث حسب الحاجة بواسطة معالج التوجيه. وباستخدام النسخ المحلية من جدول التمرير يمكن أن يتم اتخاذ قرار التمرير محلياً في كل منفذ مدخل بدون استدعاء معالج التوجيه المركزي. مثل هذه المعالجة اللامركزية تتجنب وجود اختناق (عنق زجاجة) عند نقطة واحدة داخل الموجّه.



الشكل 4-7 المعالجة بمنفذ المدخل.

تاريخ حالة (Case History)

هيمنة أنظمة سيسكو على قلب الشبكة

في أكتوبر/تشرين الأول عام 2006 (عند إعداد الكتاب الأصلي) بلغ عدد الموظفين بشركة سيسكو أكثر من 30 ألف شخصٍ، وبلغ رأسمالها في السوق حوالي 140 بليون دولاراً. تهيمن أنظمة سيسكو حالياً على سوق موجهات الإنترنت، وفي السنوات الأخيرة تحركت إلى سوق هاتف الإنترنت حيث تتنافس نداً لند مع شركات أجهزة الهاتف مثل Lucent، Alcatel، Norte، Siemens. فكيف نشأت هذه الغوريلا كشركة شبكات؟ كانت البداية في عام 1984 في غرفة المعيشة بشقّة في وادي السيليكون (Silicon Valley).

كان 'لن بوزاك' (Len Bosak) وزوجته 'ساندي لرنر' (Sandy Lerner) يعملان في جامعة ستانفورد عندما تكونت لديهم فكرة بناء وبيع موجّهات شبكة الإنترنت للمؤسسات الأكاديمية والبحثية. جاءت ساندي لرنر بالاسم سيسكو (كاختصار لـ 'سان فرانسيسكو')، كما صمّمت شعار الشركة. وكان المقر الرئيس للشركة هو غرفة المعيشة في منزلهم، وتم تمويل المشروع عن طريق بطاقات الائتمان ووظائف العمل الجزئي الاستشارية. في نهاية عام 1986 بلغت عائدات سيسكو 250 ألف دولار في الشّهر، وفي نهاية عام 1987 نجحت سيسكو في جذب رأسمال استثماري بلغ مليوني دولار من Sequoia في مقابل ثلث الشركة. وواصلت سيسكو نموها على مدى السنوات القليلة التالية وزادت حصة الشركة في السوق أكثر فأكثر.

في نفس الوقت توترت علاقات بوزاك ولرنر مع إدارة سيسكو. أشهرت سيسكو للجمهور في عام 1990، وفي نفس العام ترك بوزاك ولرنر الشركة. وعلى مرّ السنين توسّعت سيسكو أكثر لتشمل منتجات وخدمات أخرى غير الموجّهات كأجهزة الأمن واللاسلكي، ومنتجات وخدمات نقل الصوت عبر الإنترنت.

ومع ذلك واجهت سيسكو منافسة دولية متزايدة مع شركات أخرى مثل شركة Huawei وهي شركة صينية سريعة النمو. بلغ عدد موظفي شركة Huawei أكثر من 38 ألف موظفٍ حول العالم واستحوذت الشركة - حسبما أُعلن مؤخراً - على أكثر من 7٪ من أسواق الموجّهات ومحولات الإيثرنت. ومن الشركات المنافسة لسيسكو أيضاً في مجال الموجّهات ومحولات الإيثرنت شركة ألكاتيل (Alcatel) ولوسينت (Lucent) وجونيبر (Juniper).

في الموجّهات ذات قدرة المعالجة المحدودة في منفذ المدخل قد يرسل منفذ المدخل الرزمة ببساطة إلى معالج التوجيه المركزي، والذي يقوم بدوره بالبحث في جدول التمرير وإرسال الرزمة إلى منفذ المخرج المناسب. هذه الطريقة تُتبع عندما تعمل محطة عمل فرعية (workstation) أو خادم كموجّه. في تلك الحالة يكون معالج التوجيه في الحقيقة هو وحدة المعالجة المركزية CPU لمحطة العمل الفرعية، ويكون منفذ المدخل في الحقيقة هو بطاقة مواءمة الشبكة (network interface card؛ على سبيل المثال بطاقة الإيثرنت.

بوجود جدول التمرير تكون عملية البحث بسيطة من حيث المفهوم، فقط نبحث خلال جدول التمرير عن تطابق أطول بادئة، كما وصفنا في الجزء 4-2-2. ومع ذلك فليست الحياة عملياً بهذه البساطة. ربما يكون عامل الصعوبة الأول والأكثر أهمية هو أن موجّهات شبكة العمود الفقري يجب أن تعمل بسرعة عالية مؤدّيةً الملايين من عمليات البحث كل ثانية. في الحقيقة من المرغوب فيه أن تتم معالجة منفذ المدخل بسرعة الخط، أي تؤدِّي عملية البحث في زمن أقل من الوقت اللازم لاستلام رزمة في منفذ المدخل. في هذه الحالة يمكن أن تكتمل معالجة الرزمة المستلمة قبل أن تكتمل عملية الاستلام التالية. ولفهم متطلبات الأداء لعملية البحث، تصور وصلة من النوع OC-48 بسرعة 2.5 جيجابت/ثانية ورزم بحجم 256 بايت، فهذا يعني ضمناً أن سرعة البحث تبلغ تقريباً مليون عملية في الثانية.

ومع الحاجة للتشغيل بالسرعات العالية المتاحة للوصلة الآن يتضح أن البحث التتابعي (sequential search) خلال جدول تمرير كبير يصبح أمراً مستحيلاً. وهناك طريقة أخرى أكثر معقولية هي أن تُخزن مُدخلات جدول التمرير في هيكل بيانات على شكل "شجرة". حيث يمكن أن تفكر في كل مستوى في الشجرة بمثابة بتٍ في عنوان الوجهة. وللبحث عن عنوان نبدأ ببساطة من عقدة "الجذر" للشجرة ونتحرك خلال الشجرة. فإذا كان البت الأول في العنوان "0" نتجه للشجرة الفرعية اليسرى، وإذا كان "1" نتجه للشجرة الفرعية اليمنى. ثمّ نتتبع الشجرة الفرعية المناسبة باستعمال البتات الأخرى في العنوان. بهذه الطريقة يمكن أن نبحث في جدول التمرير عن عنوان في عدد من الخطوات يساوي N (وهي تمثل عدد البتات في

العنوان). لاحظ أن هذه الطريقة أساساً هي بحث ثنائي (binary search) خلال فضاء عناوين حجمه يساوي 2^N. ويمكنك الاطلاع على تعديل لطريقة البحث الثنائي في [Srinivasan 1999] وعلى دراسة مسحية عامة لخوارزميات تصنيف الرزم في [Gupta 2001].

لكن حتى مع $N = 32$ (على سبيل المثال عنوان IP يتكون من 32 بتاً) لا تكون سرعة البحث بأسلوب "البحث الثنائي" سريعة بما فيه الكفاية لمتطلبات توجيه شبكة العمود الفقري اليوم. على سبيل المثال افترض أن كل خطوة تحتاج إلى وصول للذاكرة (memory access)، فإن أقل من مليون بحث في الثانية يمكن أن يتم إذا كانت سرعة الوصول للذاكرة 40 نانو ثانية (40ns). وعليه تم اقتراح عدّة أساليب لزيادة سرعة البحث. على سبيل المثال تسمح ذاكرة CAM – وهي ذاكرة معنونة بمحتوياتها (content addressable memory) – باستخراج محتوى مُدخل جدول التمرير المناظر لعنوان IP المقدّم للذاكرة في وقت ثابت بالضرورة. وتُعتبر سلسلة سيسكو 8500 مثالاً للموجهات التي تحتوي على ذاكرة CAM حيث توجد ذاكرة من هذا النوع بسعة 64 كيلوبايت (64KB) لكل منفذ مدخل.

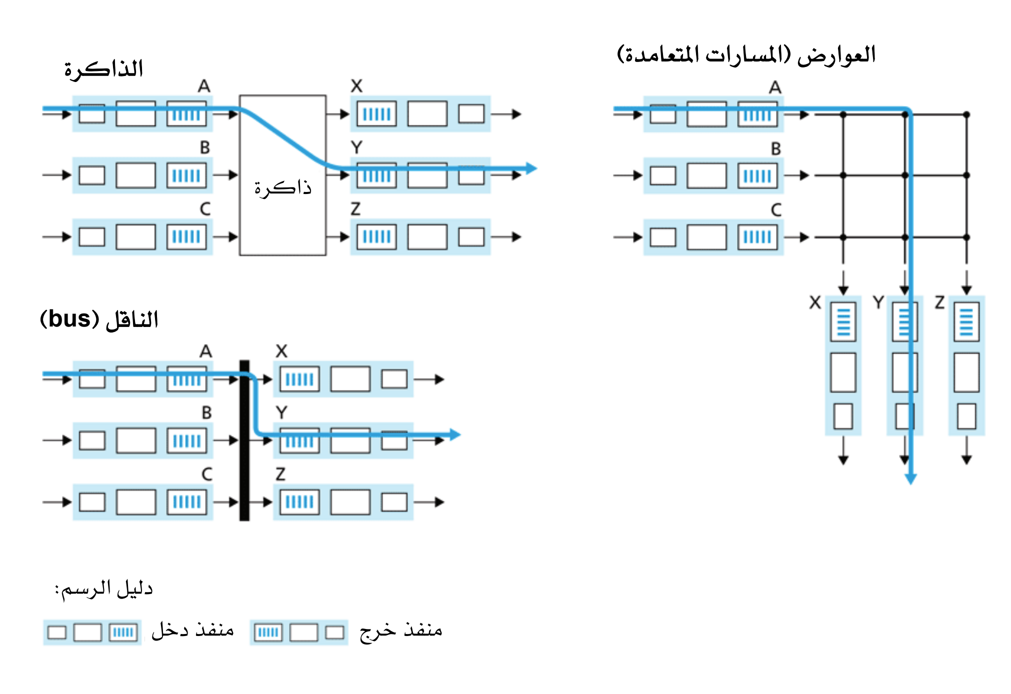
طريقة أخرى لتسريع البحث هي الاحتفاظ بمُدخلات جدول التمرير التي استُخدمت مؤخراً في ذاكرة وسيطة (cache) [Feldmeier 1988]. لكن تبقى مشكلة الحجم المحتمل للذاكرة الوسيطة. اقتُرح أيضاً تراكيب (هياكل) بيانات سريعة تسمح بالبحث عن مُدخلات جدول التمرير في عدد خطوات يساوي $\log(N)$ [Waldvogel 1997] وأساليب أخرى تقوم بضغط جداول التمرير بطرق مبتكرة [Brodnik 1997]. كما تم مناقشة طريقة "مادية" (hardware approach) محسّنة للبحث تفيد في الحالة الشائعة عندما يكون العنوان المراد البحث عنه يتضمن 24 بتاً أو أقل [Gupta 1998]. للحصول على دراسة مسحية وتصنيف لأنواع الخوارزميات السريعة للبحث داخل جداول التمرير اطّلِع على [Ruiz Sanchez 2001].

وبمجرد تحديد منفذ المخرج للرزمة عن طريق البحث يمكن أن تُرسَل الرزمة خلال نسيج التحويل. لكن قد تُمنع الرزمة بشكلٍ مؤقت من دخول نسيج التحويل

(بسبب انشغال النسيج حالياً لنقل رزم من منافذ المدخل الأخرى). لذا يجب أن تنتظر الرزمة المستوقفة في منفذ المدخل وبعد ذلك تُجدول (scheduled) لعبور نسيج التحويل في وقتٍ لاحق. سنلقي نظرة أكثر تفحصاً على الإيقاف (blocking) والانتظار في الصف (الطابور) (queueing) وجدولة الرزم (في منافذ المدخل ومنافذ المخرج) داخل موجّه في الجزء 4-3-4.

4-3-2 نسيج التحويل

يوجد نسيج التحويل في صميم قلب الموجّه. فمن خلال نسيج التحويل تنتقل الرزم في الحقيقة (أي تُمرر) من منفذ مدخل إلى منفذ مخرج. يمكن أن ينجز التحويل بعدة طرق كما هو مبين في الشكل 4-8:



الشكل 4-8 ثلاثة طرق للتحويل.

- التحويل عن طريق الذاكرة: كانت الموجّهات الأولى البسيطة حاسبات تقليدية في أغلب الأحيان، وكان التحويل يتم بين منافذ المدخل ومنافذ المخرج تحت السيطرة المباشرة لوحدة المعالجة المركزية (معالج التوجيه). وكانت منافذ المدخل والمخرج تمثل أجهزة إدخال وإخراج تقليدية في نظام التشغيل التقليدي. عند وصول رزمة إلى منفذ مدخل يبعث "إشارة مقاطعة" (interrupt) إلى معالج التوجيه. بعد ذلك تُنسَخ الرزمة من منفذ المدخل إلى ذاكرة المعالج، حيث يقوم معالج التوجيه باستخلاص عنوان الوجهة من ترويسة الرزمة، والبحث عن منفذ المخرج المناسب في جدول التمرير، ثم نسخ الرزمة إلى الذاكرة المؤقتة (المرحلية) (buffer) لمنفذ المخرج. لاحظ أنه إذا كان الحيز الترددي للذاكرة بحيث يمكن الكتابة فيها أو القراءة منها بسرعة B رزمة في الثانية، فإن الطاقة الإنتاجية الكلية للتمرير (المعدل الكلي لنقل الرزم من منافذ المدخل إلى منافذ المخرج) يجب أن تكون أقل من $B/2$.

  يتم التحويل أيضاً في العديد من الموجّهات الحديثة عن طريق الذاكرة. ومع ذلك هناك اختلاف أساسي عن الموجّهات المبكرة، وهو أن البحث عن عنوان الوجهة وتخزين الرزمة في موقع الذاكرة المناسب يتم بواسطة المعالجات على بطاقات الخط للإدخال. وتشبه الموجّهات التي تحول الرزم عن طريق الذاكرة إلى حدٍ كبير المعالجات المتعددة ذات الذاكرة المشتركة (shared-memory multiprocessors)، حيث تقوم المعالجات على بطاقة الخط بتحويل الرزم إلى ذاكرة منفذ المخرج المناسب. وكمثال تقوم سلسلة محوّلات Cisco Catalyst 8500 [Cisco 8500 2007] بتحويل الرزم عن طريق الذاكرة المشتركة. ويمكنك الاطلاع على نموذج تجريدي لدراسة خواص التحويل المبني على الذاكرة ومقارنته بالأشكال الأخرى من التحويل في [Iyer 2002].

- التحويل عن طريق "ناقل" (bus): في هذه الطريقة تنقل منافذ المدخل الرزمة مباشرةً إلى منفذ المخرج على ناقل مشترك بدون تدخُّل من معالج التوجيه

(لاحظ أنه عند التحويل عن طريق الذاكرة يجب أيضاً أن تعبر الرزمة ناقل النظام إلي الذاكرة أو منها). وبالرغم من أن معالج التوجيه لم يشترك في النقل على الناقل إلا أنه يمكن أن تنتقل رزمة واحدة فقط في كل مرة على الناقل (لأن الناقل مشترك). عندما تصل رزمة إلى منفذ مدخل وتجد الناقل مشغولاً بنقل رزمة أخرى فسوف تُمنع من عبور نسيج التحويل وعليها الانتظار في طابور منفذ المدخل. ولأن كل رزمة يجب أن تعبر الناقل الوحيد فإن سعة التحويل (switching bandwidth) للموجّه محدودة بسرعة الناقل.

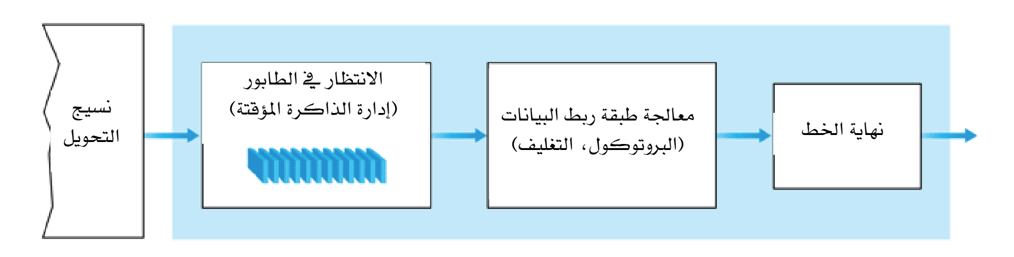
ومع توفر سعات للناقل في تقنيات اليوم تزيد على جيجابت في الثانية، فإن التحويل عن طريق الناقل يعتبر كافياً في أغلب الأحيان لموجّهات شبكات الوصول وشبكات المؤسسات (كالشبكات المحلية وشبكات الشركات). ويُستخدم التحويل عن طريق الناقل في عدد من منتجات الموجّهات الحالية مثل Cisco 5600 [Cisco Switches 2007] والتي تحوّل الرزم على ناقل لوحات الربط الخلفية (backplane bus) بسرعات تزيد على 32 جيجابت/ثانية.

- التحويل عن طريق شبكة ربط بينية (interconnection network): أحد الطرق للتغلب على قيود الحيز الترددي لناقل مشترك وحيد هو استخدام شبكة ربط بيني أكثر تطوّراً كتلك التي استعملت في الماضي لربط المعالجات في البنية المعمارية للحاسب متعدد المعالجات. يمثل محوّل العوارض (محوّل بمسارات متعامدة) (crossbar switch) شبكة ربط بيني تتكون من ناقلات عددها $2n$ لتوصل n منفذ مدخل إلى n منفذ مخرج كما هو مبين في الشكل 4-8. تنتقل الرزمة التي تصل إلى منفذ مدخل على الناقل الأفقي المتصل بمنفذ المدخل حتى يتقاطع بالناقل العمودي المؤدّي إلى منفذ المخرج المطلوب. إذا كان الناقل العمودي المؤدّي إلى منفذ المخرج حراً ("غير مشغول") تنتقل الرزمة إلى منفذ المخرج. أما إذا كان الناقل العمودي مستخدماً لنقل رزمة من منفذ مدخل آخر إلى نفس منفذ المخرج فإن تلك الرزمة تُمنَع ويجب عليها أن تنتظر في طابور منفذ المدخل.

اقتُرحت أيضاً أنسجة تحويل الدلتا والأوميجا (Delta and Omega switching fabrics) كشبكة ربط بين منافذ المدخل والمخرج. انظر [Tobagi 1990] لدراسة مسحية للبنى المعمارية للمحوّلات. وكمثال، تُستعمل محوّلات عائلة سيسكو 12000 [Cisco 12000 2007] شبكة ربط بسرعات تصل إلى 60 جيجابت/ثانية خلال نسيج التحويل. أحد الاتجاهات في تصميم شبكة الربط [Keshav 1998] هو تجزئة رزمة IP ذات الطول المتغير إلى خلايا ثابتة الطول، ثم تُعلَّم وتُنقَل تلك الخلايا خلال شبكة الربط. يعاد تجميع الخلايا إلى الرزمة الأصلية في منفذ المخرج. يمكن أن يبسّط استخدام الخلايا ثابتة الطول والتعليم الداخلي عملية تحويل الرزم خلال شبكة الربط إلى حدٍّ كبير ويزيد من سرعتها.

4-3-3 منافذ المخرج

تتضمن المعالجة – التي تتم في منافذ المخرج والموضحة في الشكل 4-9 – أخذ الرزم التي خُزنت في ذاكرة منفذ المخرج وإرسالها على الوصلة الخارجة. تمثل معالجة بروتوكول وصلة البيانات وتوفير نهاية للخط وظائف طبقة ربط البيانات والطبقة المادية لجهة الإرسال التي تتفاعل مع منفذ المدخل على الطرف الآخر للوصلة الخارجة، كما نوقش في الجزء 4-3-1. إن وظائف إدارة الانتظار في الصف وإدارة الذاكرة المؤقتة (buffer) مطلوبة عندما يُسلِّم نسيج التحويل الرزم إلى منفذ المخرج بمعدل يتجاوز معدل الإرسال على وصلة المخرج، وسوف نغطي الانتظار في صفوف منافذ المخرج فيما بعد.



الشكل 4-9 المعالجة في منفذ المخرج.

4-3-4 أين يحدث الانتظار في الطابور؟

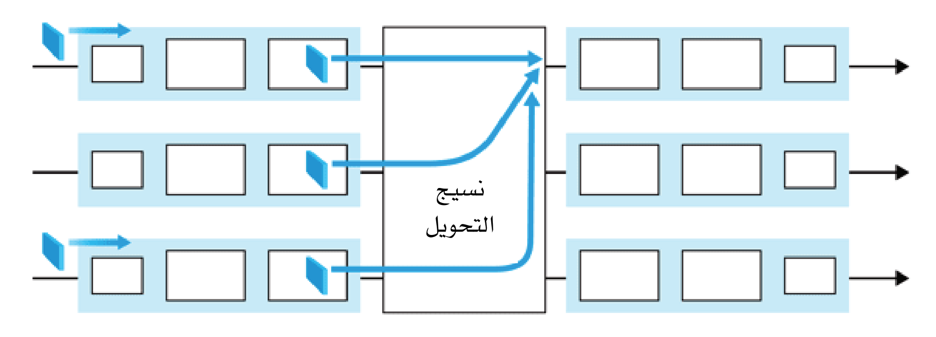
إذا نظرنا إلى وظائف منافذ المدخل والمخرج والترتيبات المبينة في الشكل 4-8 يتضح أن طوابير الرزم يمكن أن تُشكَّل في كلٍّ من منافذ المدخل ومنافذ المخرج. من المهم دراسة هذه الطوابير بشيء من التفصيل لأنه مع زيادة حجمها ستُستنزَف ذاكرة الموجّه المؤقتة في النهاية، وسوف يؤدي ذلك إلى فقد الرزم. تذكّر أننا ذكرنا في مناقشاتنا السابقة أن الرزم تُفقَد داخل الشبكة أو تُسقَط عند الموجّه. وها نحن نرى كيف تُسقَط مثل تلك الرزم وتُفقَد هنا في هذه الطوابير داخل الموجّه. يعتمد الموقع الفعلي لفقد الرزم (في طوابير منفذ المدخل أو طوابير منفذ المخرج) – كما سنناقش فيما بعد – على حِمْل مرور البيانات (traffic load) والسرعة النسبية لنسيج التحويل وسرعة الخط.

افترض أن سرعة الخط لكلٍّ من منافذ المدخل والمخرج متماثلة، وأن هناك عدد n منفذ مدخل وعدد n منفذ مخرج. ولنُعرّف سرعة نسيج التحويل على أنها المعدل الذي يمكن به لنسيج التحويل أن يحرّك الرزم من منافذ المدخل إلى منافذ المخرج. إذا كانت سرعة نسيج التحويل تعادل على الأقل n مرة سرعة خط المدخل فعندئذ لا يمكن أن يحدث أي انتظار في الطوابير في منافذ المدخل. وذلك لأنه حتى في أسوأ الأحوال عندما تستلم كل منافذ المدخل رزماً فلا يزال بوسع المحوّل نقل عدد n رزمة من منفذ المدخل إلى منفذ المخرج في نفس الوقت الذي يأخذه كل منفذ من منافذ المدخل (بشكلٍ آني) لاستلام رزمة واحدة.

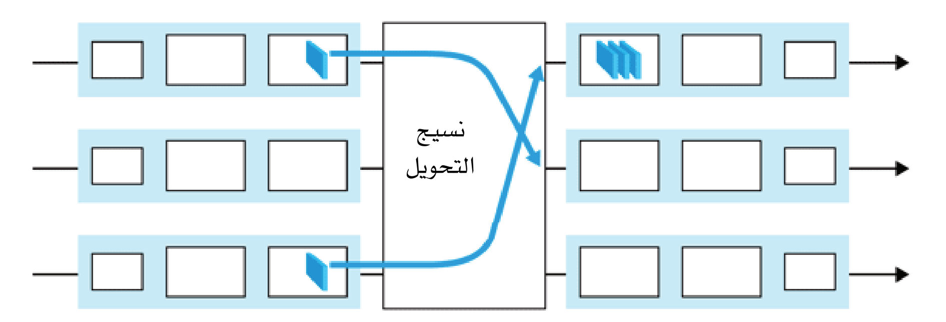
لكن ماذا يمكن أن يحدث في منافذ المخرج؟ دعنا نفترض بأنه ما زالت سرعة نسيج التحويل تعادل على الأقل n مرة سرعة الخط. في أسوأ الأحوال تكون كل الرزم التي تصل إلى منافذ المدخل (وعددها n) متجهة إلى نفس منفذ المخرج. في هذه الحالة في خلال الوقت اللازم لاستلام (أو إرسال) رزمة واحدة ستصل n رزمة إلى منفذ المخرج هذا. ولأنه يمكن أن يرسل منفذ المخرج رزمة واحدة فقط في وحدة الزمن (وقت إرسال الرزمة) فعندئذ سيكون على الرزم المُستلَمة (وعددها n) أن تصطف (تنتظر) للإرسال على الوصلة الخارجة. وسيكون من المحتمل وصول رزم

أكثر عددها n رزمة في الوقت اللازم لإرسال رزمة واحدة من تلك التي سبق وضعها في الطابور. وهكذا في النهاية يمكن أن يزيد عدد الرزم المنتظرة بدرجة كافية لاستنزاف الذاكرة المؤقتة في منفذ المخرج وعند ذلك يبدأ إسقاط (فقد) الرزم.

التنازع على منفذ المخرج عند الزمن t



بعد زمن رزمة واحدة



الشكل 4-10 الانتظار في منافذ المخرج.

يوضح الشكل 4-10 الانتظار في طوابير منافذ المخرج. عند زمن t تصل رزمة إلى كل منفذ من منافذ المدخل، وكلٌّ منها متجه إلى منفذ المخرج الموجود في أعلى الشكل. افترض أن سرعة الخط متماثلة وأن المحوّل يعمل بسرعة تعادل ثلاثة أضعاف سرعة الخط. بعد وحدة زمن (أي الوقت اللازم لاستلام أو إرسال رزمة) تكون الرزم الثلاثة الأصلية قد نُقلت إلى منفذ المخرج واصطفت منتظرة الإرسال.

في وحدة الزمن التالية سترسل إحدى هذه الرزم الثلاثة على الوصلة الخارجة. في مثالنا ستصل رزمتان جديدتان إلى الجانب القادم للمحوّل؛ إحداهما متجهة إلى نفس منفذ المخرج الموجود في أعلى الشكل.

بافتراض أن الذاكرة المؤقتة لازمة لامتصاص التقلبات في حِمْل مرور البيانات، فالسؤال الطبيعي الآن هو "ما الحجم المطلوب لتلك الذاكرة؟". لعدة سنوات كانت القاعدة التقريبية الشائعة (المبنية على التجربة العملية وليس المعرفة العلمية) [RFC 3439] لاختيار حجم الذاكرة المؤقتة B هي أن يكون مساوياً لحاصل ضرب متوسط زمن الرحلة ذهاباً وإياباً (RTT مثلا 250 ميلي ثانية) وسعة الوصلة C. وهذه النتيجة مستندة على تحليل ديناميكا الطوابير لعدد صغير نسبياً من مسارات تدفق TCP [Villamizar 1994]. فمثلاً إذا كانت سعة الوصلة 10 جيجابت في الثانية وقيمة RTT تساوي 250 ميلي ثانية فإن حجم الذاكرة المؤقتة التي نحتاجها $B=RTT\times C=2.5$Gb. وتقترح الجهود النظرية والتجريبية الحديثة حساب حجم الذاكرة عندما يكون هناك عدد كبير من مسارات تدفق TCP يمر بالوصلة من المعادلة $B = RTT \times C/\sqrt{N}$ حيث تمثل N عدد تلك المسارات [Appenzeller 2004]. ومع وجود عدد كبير من مسارات التدفق يمر خلال وصلات شبكة عمود فقري كبيرة (انظر على سبيل المثال [Fraleigh 2003]) يمكن أن تكون قيمة N كبيرة جداً، وبالتالي يقل حجم الذاكرة المطلوبة بشكلٍ ملحوظ للغاية. قدَّم [Appenzeller 2004] و [Wischik 2005] مناقشات سهلة القراءة لمشكلة اختيار حجم الذاكرة المؤقتة من منطلقات نظرية وتطبيقية وتشغيلية.

نتيجة للانتظار في طابور منفذ المخرج يجب أن يختار مُجدول الرزم (packet scheduler) في منفذ المخرج رزمة واحدة من بين تلك الرزم المنتظرة للإرسال. قد يتم هذا الاختيار بقاعدة بسيطة مثل قاعدة "الأول وصولاً ... الأول خدمةً" (FCFS) والتي تعطي أفضلية الخدمة للأول وصولاً، أو نظام جدولة أكثر تطوّراً مثل قاعدة طوابير الانتظار العادلة ذات الأوزان (WFQ)، والتي يتم فيها تقاسم الوصلة الخارجة بإنصاف بين التوصيلات المختلفة من طرف لطرف والتي لها رزم منتظرة للإرسال. وتلعب جدولة الرزم دوراً هاماً لتوفير ضمانات جودة للخدمة (quality of service).

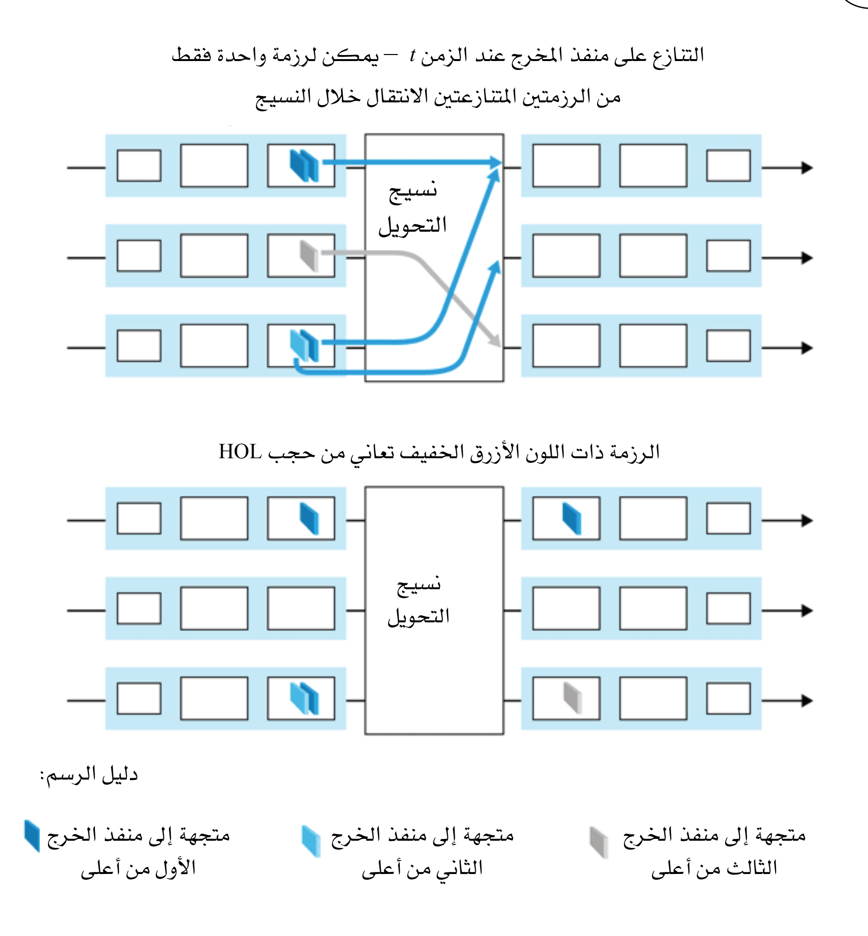
سوف نغطي جدولة الرزم بتوسع أكثر في الفصل السابع، وهناك مناقشة حول قواعد جدولة الرزم لمنافذ المخرج في [Cisco Queue 2007].

بنفس الطريقة إذا لم تكن هناك ذاكرة مؤقتة تكفي لتخزين الرزمة القادمة يجب أن يُتخَذ قرار إما بإسقاط الرزمة المستلمة (وهي سياسة تعرف بالإسقاط الذيلي) أو لإزالة واحدة أو أكثر من الرزم المنتظرة في الطابور لإخلاء مكان للرزمة الواصلة حديثاً. وقد يكون من المفيد في بعض الحالات إسقاط رزمة (أو التأشير على ترويستها) قبل امتلاء الذاكرة المؤقتة لكي تعطي إشارة ازدحام إلى المُرسِل. تم اقتراح وتحليل عدة سياسات لإسقاط الرزم والتأشير عليها، وأصبحت تعرف مجتمعةً بخوارزميات إدارة الطابور النشطة (Active Queue Management (AQM)) [Labrador 1999; Hollot 2002]. يطلق على إحدى هذه الخوارزميات المدروسة والمطبّقة على نحو واسع خوارزمية "الكشف المبكّر العشوائي" (Random-Early Detection (RED))، وبهذه الطريقة يمكن الاحتفاظ بمتوسط موزون (weighted average) لطول طابور المخرج. إذا كان طول الطابور المتوسط أقل من عتبة الحد الأدنى (minimum threshold) min_{th} فعندما تصل رزمة سوف يسمح لها بالانتظار في الطابور. وبالمقابل إذا كان الطابور ممتلئاً بالكامل أو أن طول الطابور المتوسط أعلى من عتبة الحد الأقصى max_{th} فعندما تصل رزمة سوف يُؤشَّر عليها أو تُسقَط. وأخيراً إذا وصلت الرزمة وكان طول الطابور المتوسط في المدى [min_{th}, max_{th}] سوف يُؤشَّر عليها أو تُسقَط باحتمالية معينة والتي عادة ما تكون دالة في (أي تعتمد على) طول الطابور المتوسط وmin_{th} وmax_{th}. اقتُرحت عدة طرق للتأشير والإسقاط الاحتمالي، وتم نمذجة وتحليل ومحاكاة وتطبيق نسخ مختلفة من طريقة RED. قدَّم [Christiansen 2001] و[Floyd 2007] نظرة عامة حول هذا الموضوع مع ذكر مراجع للقراءة الإضافية.

إذا كان نسيج المحوّل ليس سريعاً بما فيه الكفاية (مقارنةً بسرعة خط المدخل) لتحويل كل الرزم الواصلة خلال النسيج بدون تأخير فإن الانتظار يمكن أن يحدث أيضاً في منافذ المدخل؛ لأن الرزم يجب أن تلتحق بطوابير منفذ المدخل لانتظار دورها قبل أن تنتقل خلال نسيج التحويل إلى منفذ المخرج. ولتوضيح نتيجة

مهمة لهذا الانتظار افترض وجود نسيج تحويل بمسارات متعامدة (crossbar switching fabric) وأن (1) كل سرعات الوصلات متماثلة، (2) زمن تحويل رزمة واحدة من أي منفذ مدخل إلى منفذ مخرج معين هو نفسه الزمن الذي تأخذه رزمة لاستلامها على منفذ مدخل، (3) نقل الرزم من طابور منفذ المدخل إلى طابور المخرج المطلوب يتم بأسلوب FCFS. يمكن أن تحوّل رزم متعدّدة بالتوازي (في نفس الوقت) طالما أن منافذ المخرج مختلفة. لكن إذا كانت رزمتان في مقدمة طابورَي دخل تتّجهان إلى نفس طابور المخرج فإن إحديهما ستوقف ويجب أن تنتظر في طابور المدخل (لأن نسيج التحويل يمكن أن يحوّل رزمة واحدة فقط إلى منفذ مخرج معين في وقتٍ ما).

يبين الشكل 4-11 مثالاً فيه رزمتان (مظلّلتان على نحو داكن) في مقدمة طابورَي دخل ومتجهتان إلى نفس منفذ المخرج الموجود أعلى الشكل. افترض أن نسيج المحوّل يختار تحويل الرزمة من مقدمة الطابور الموجود في أعلى اليسار. في هذه الحالة يجب أن تنتظر الرزمة المظلّلة على نحو داكن في الطابور الموجود أسفل اليسار. وليس هذا فحسب بل يجب أيضاً أن تنتظر الرزمة المظلّلة قليلاً والموجودة وراء تلك الرزمة في الطابور أسفل اليسار بالرغم من عدم وجود تنازع على منفذ المخرج في وسط اليمين (والذي يمثل وجهة الرزمة المظلّلة قليلاً). تعرف هذه الظاهرة بـ "حجب مقدمة الطابور (HOL blocking)" في المحوّلات ذات الطوابير عند منافذ المدخل (أي يجب أن تنتظر الرزمة الموجودة في طابور منفذ مدخل بالرغم من أن منفذ المخرج لها قد يكون حراً وذلك نظراً لوجود رزمة أخرى منتظرة في مقدمة الطابور). بيّن [Karol 1987] أنه – تحت بعض الفرضيات – تؤدي ظاهرة حجب HOL إلى زيادة طول طابور المدخل بطريقة غير محدودة (أي يحدث فقدٌ ملحوظٌ في الرزم) بمجرد أن يصبح معدل وصول الرزم على وصلات المدخل 58٪ فقط من سعتها. وتوجد مناقشة لعدد من الحلول لمشكلة "حجب HOL" في [McKeown 1997b].



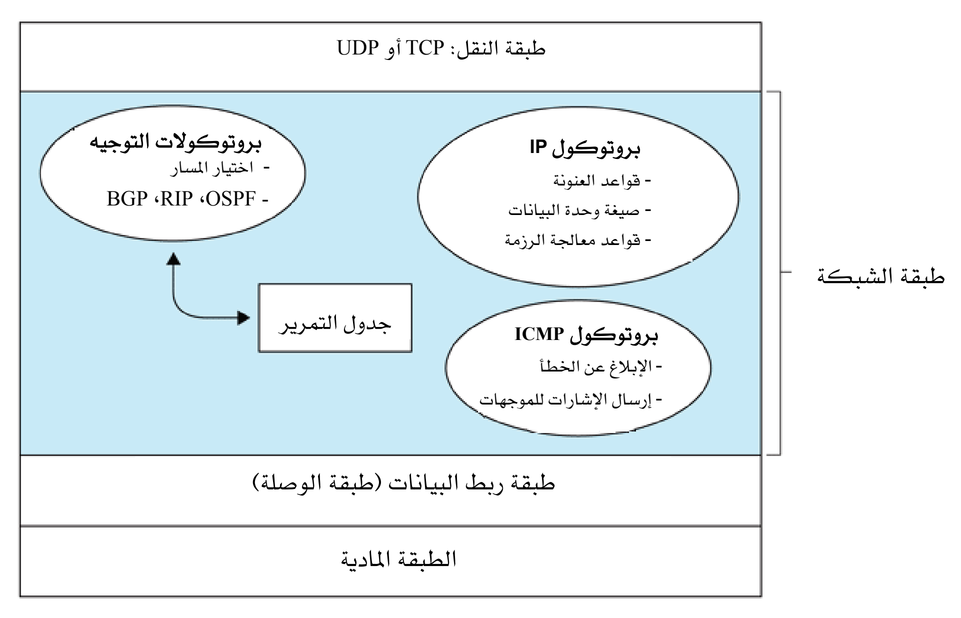
الشكل 4-11 حجب مقدمة الطابور.

4-4 بروتوكول الإنترنت (IP): التمرير والعنونة في الإنترنت

لقد كانت مناقشتنا للعنونة والتمرير في طبقة الشبكة حتى الآن عامة وغير مرتبطة بشبكة محددة. في هذا الجزء سوف نحوّل انتباهنا لكيفية إنجاز العنونة والتمرير في شبكة الإنترنت بصفة خاصةً. وسنرى بأن العنونة والتمرير في الإنترنت تعد مكوّنات هامة في بروتوكول الإنترنت IP. توجد نسختان من بروتوكول الإنترنت IP قيد الاستعمال اليوم. سوف نفحص أولاً نسخة بروتوكول الإنترنت 4 الواسعة الانتشار والتي عادةً ما يشار إليها بـ IPv4 [RFC 791]. وسوف نفحص نسخة

بروتوكول الإنترنت 6 في نهاية هذا الجزء [RFC 2460; RFC 3513] (والتي اقتُرِحَت لتحل محل IPv4).

لكن قبل أن نبدأ حملتنا لاستكشاف بروتوكول الإنترنت دعنا نراجع مكوّنات طبقة شبكة الإنترنت. كما هو موضح في الشكل 4-12 تتكون طبقة شبكة الإنترنت من ثلاثة مكوّنات رئيسة. الأول بروتوكول الإنترنت (موضوع هذا الجزء)، والثاني التوجيه لتحديد المسار الذي تتبعه رزمة البيانات من مصدرها إلى وجهتها. ذكرنا في وقت سابق أن بروتوكولات التوجيه تحسب جداول التمرير التي تستعمل لإرسال الرزم خلال الشبكة. وسوف ندرس بروتوكولات التوجيه في الإنترنت في الجزء 4-6. أما المكوّن النهائي لطبقة الشبكة فهو وسيلة للإبلاغ عن الأخطاء في رزم البيانات والرد على الطلبات لمعلومات معينة من طبقة الشبكة. وسوف نغطي بروتوكول الإبلاغ عن الأخطاء والمعلومات في طبقة شبكة الإنترنت والمشار إليه في الجزء 4-4-3 ببروتوكول رسائل التحكم في الإنترنت (Internet Control Message Protocol (ICMP)).



الشكل 4-12 طبقة الشبكة في الإنترنت.

4-4-1 صيغة وحدة البيانات

تذكّر أن رزمة طبقة الشبكة تُدعى وحدة بيانات. سنبدأ دراستنا لبروتوكول الإنترنت بنظرة عامة لتراكيب (دراسة القواعد النحوية) ودلالات (دراسة المعاني) وحدة بيانات بروتوكول IPv4. قد تعتقد أنه لا شيء يمكن أن يكون أكثر جفافاً من دراسة النحو ودراسة معاني البتات في الحقول المختلفة للرزمة. على الرغم من ذلك تلعب وحدة البيانات دوراً هاماً في الإنترنت (فكل دارس ومحترف للشبكات يحتاج لأن يراها ويستوعبها ويتقنها). يوضح الشكل 4-13 صيغة وحدة بيانات IPv4. إن الحقول الرئيسة في وحدة بيانات IPv4 كالتالي:

32 بتاً

طول وحدة البيانات (عدد البايتات)	نوع الخدمة	طول الترويسة	رقم الإصدار
عنوان التجزئة (13 بتاً)	أعلام	معرّف (16 بتاً)	
المجموع التدقيقي للترويسة	بروتوكول الطبقة العليا	فترة العمر	
عنوان المصدر (32 بتاً)			
عنوان الوجهة (32 بتاً)			
الخيارات (إذا وجدت)			
البيانات			

الشكل 4-13 صيغة وحدة بيانات بروتوكول IPv4.

- رقم النسخة: تحدّد الـ 4 بتات الأولى رقم نسخة بروتوكول الإنترنت لوحدة البيانات تلك. بالنظر إلى رقم النسخة يستطيع الموجّه أن يحدّد كيف يترجم بقيّة حقول وحدة البيانات. تستعمل النسخ المختلفة لبروتوكول الإنترنت صيغ

وحدات بيانات مختلفة. يوضح الشكل 4-13 صيغة وحدة البيانات للنسخة الحالية لبروتوكول الإنترنت IPv4. سوف نناقش صيغة وحدة البيانات للنسخة الجديدة لبروتوكول الإنترنت IPv6 في نهاية هذا الجزء.

- طول الترويسة: لأن وحدة بيانات IPv4 يمكن أن تحتوي على عدد متغير من الخيارات (التي تُتضمّن في ترويسة وحدة البيانات) فهذه البتات الأربعة مطلوبة لتقرير أين تبدأ البيانات فعلاً في وحدة البيانات. معظم وحدات بيانات بروتوكول الإنترنت لا تحتوي على خيارات في الترويسة، ولذا فإن ترويسة وحدة بيانات بروتوكول الإنترنت العادية تتكون من 20 بايتاً.
- نوع (نمط) الخدمة: تشتمل ترويسة IPv4 على حقل نوع الخدمة (TOS) للسماح بتمييز أنواع مختلفة من وحدات البيانات عن بعضها البعض (مثلاً رزم بيانات تتطلّب بصفة خاصة تأخيراً منخفضاً أو طاقة إنتاجية عالية أو موثوقية نقل). على سبيل المثال قد يكون من المفيد تمييز وحدات البيانات الفورية (كتلك المستخدمة من قِبَل تطبيق هاتف الإنترنت) عن غيرها (على سبيل المثال FTP). يعتبر المستوى المعين للخدمة الذي يمكن توفيره قضية سياسة تُحدَّد من قِبَل مدير الموجّه. سوف نستكشف موضوع الخدمة التفاضلية (differentiated service) بالتفصيل في الفصل السابع.
- طول وحدة البيانات: وهو يمثل الطول الكلي لوحدة بيانات بروتوكول الإنترنت (الترويسة والبيانات) مقاسة بالبايتات. ولأن هذا الحقل طوله 16 بتاً، فإن الحجم الأقصى النظري لوحدة بيانات بروتوكول الإنترنت هو 65535 بايتاً. لكن من النادر أن تكون وحدات البيانات أكبر من 1500 بايت.
- المُعرّف (مُميّز الرزمة) والأعلام والعنوان النسبى للتجزئة: تُستخدم هذه الحقول الثلاثة مع ما يسمّى بـ "التجزئة" (fragmentation)، وهو موضوع سوف ندرسه بتعمّق بعد قليل. وبشكلٍ مثيرٍ للانتباه لا تسمح النسخة الجديدة لبروتوكول الإنترنت IPv6 بالتجزئة في الموجّهات.
- فترة العُمر ((Time-To-Live (TTL)): يستعمل هذا الحقل لضمان أن وحدة البيانات لا تظل تدور إلى الأبد خلال الشبكة (على سبيل المثال بسبب وجود حلقة توجيه طويلة الأمد (long-lived routing loop). تخفض قيمة هذا الحقل

بمقدار واحد في كل مرة تُعالج وحدة البيانات بموجّه. ويجب أن تُسقَط وحدة البيانات إذا أصبحت قيمة الحقل تساوى صفراً.

- البروتوكول: يُستعمَل هذا الحقل فقط عندما تصل وحدة البيانات إلى الوجهة حيث تشير قيمته إلى البروتوكول المحدد في طبقة النقل الذي يجب أن يعبر إليه هذا الجزء من وحدة البيانات. على سبيل المثال القيمة 6 تدل على أن هذا الجزء من وحدة البيانات يسلم إلى TCP، بينما القيمة 17 تدل على أنه يسلم إلى UDP. وللاطلاع على قائمة بكل القيم المحتملة راجع [RFC 1700; RFC 3232]. لاحظ أن رقم البروتوكول في وحدة بيانات IP له دور مماثل لدور حقل رقم المنفذ في قطعة بيانات طبقة النقل (segment). ويعتبر رقم البروتوكول الصمغ الذي يربط طبقة الشبكة وطبقة النقل سويةً، في حين يعتبر رقم المنفذ الصمغ الذي يربط طبقة النقل مع طبقة التطبيقات سويةً. سنرى في الفصل الخامس أن إطار طبقة ربط البيانات له أيضاً حقل خاص يربط طبقة ربط البيانات بطبقة الشبكة.
- المجموع التدقيقي للترويسة (header checksum): يساعد هذا الحقل الموجّه في اكتشاف حدوث خطأ في وحدة البيانات المستلمة. يُحسَب المجموع التدقيقي بمعاملة كل بايتين في الترويسة كعدد، ثم تجمع الأعداد الناتجة بطريقة حساب مكمل الواحد (1's complement arithmetic). وكما ناقشنا في الجزء 3-3 يُعرَف مكمل الواحد لهذا المجموع باسم المجموع التدقيقي للإنترنت ويخزَّن في حقل المجموع التدقيقي للرزمة. ويحسب الموجّه المجموع التدقيقي للترويسة لكل وحدة بيانات مستلمة، فإذا كانت القيمة المحسوبة لا تساوى القيمة المتضمنة في وحدة البيانات فإن هذا يدل على حدوث خطأ بها. وعادةً ما تُسقِط الموجّهات وحدات البيانات التي يُكتشَف وجود خطأ فيها. لاحظ أنه يجب أن تُحسَب قيمة المجموع التدقيقي عند كل موجّه ويعاد تخزينها في وحدة البيانات لأن بعض الحقول قد تتغيّر مثل حقل TTL ومن المحتمل حقل الخيارات أيضاً. يمكنك الاطلاع على مناقشة هامة لخوارزميات سريعة لحساب المجموع التدقيقي للإنترنت في [RFC 1071]. وعادة ما يُطرَح سؤال عند تلك النقطة "لماذا يقوم نموذج TCP/IP بفحص

الأخطاء في كلٍّ من طبقة النقل وطبقة الشبكة؟". هناك عدّة أسباب لهذا التكرار. أولاً: لاحظ أنه في طبقة الشبكة يتم حساب المجموع التدقيقي للترويسة فقط، بينما في طبقة النقل يتم حساب المجموع التدقيقي لقطعة البيانات بكاملها. ثانياً: بروتوكول IP وبروتوكول TCP أو UDP لا ينتميان بالضرورة لنفس رصّة البروتوكولات. فمن حيث المبدأ يمكن أن يعمل بروتوكول TCP على بروتوكول مختلف عن IP (مثلاً ATM) وكذلك بروتوكول IP يمكن أن يحمل بيانات غير متجهة إلى أيٍّ من TCP أو UDP.

- عنوان IP للمصدر وعنوان IP للوجهة: عندما يُنشئ المصدر وحدة بيانات فإنه يضع عنوانه في حقل عنوان IP للمصدر ويضع عنوان الوجهة في حقل عنوان IP للوجهة. ويحصل مضيف المصدر على عنوان الوجهة غالباً عن طريق بحث DNS كما نوقش في الفصل الثاني. سوف نناقش عنونة بروتوكول الإنترنت بالتفصيل في الجزء 4-4-2.
- الخيارات: تسمح حقول الخيارات لترويسة وحدة بيانات IP بالتمدد. غير أن خيارات الترويسة نادراً ما تستعمل، لذا كان القرار بعدم تضمين البيانات الموجودة في الحقول الاختيارية بصفة ثابتة في كل وحدة بيانات وذلك لتقليل العبء الإضافي (overhead). ومع ذلك فمجرّد وجود تلك الحقول يعقّد الأمور، فتغيير طول ترويسة وحدة البيانات يعوق إمكانية تحديد مكان بداية حقل البيانات مسبقاً. أيضاً قد تتطلّب بعض وحدات البيانات معالجة الخيارات في حين لا تحتاج بعضها الآخر لذلك، وبالتالي يمكن أن يتفاوت مقدار الوقت اللازم لمعالجة وحدة البيانات في الموجّه تفاوتاً كبيراً. هذه الاعتبارات مهمة جداً لمعالجة IP في الموجّهات والمضيفات ذات الأداء العالي. لهذه الأسباب وأسباب أخرى لم تُستخدَم خيارات IP في ترويسة IPv6 كما سنناقش في الجزء 4-4-4.
- البيانات (الحمل الآجر): نأتي أخيراً إلى الحقل الأخير والأكثر أهمية فهو المبرر الأساسي لرزمة البيانات! يحتوي هذا الحقل في أغلب الأحيان على قطعة بيانات طبقة النقل (من بروتوكول TCP أو UDP) المطلوب تسليمها إلى

وجهتها. ومع ذلك يمكن أن يحمل هذا الحقل أنواعاً أخرى من البيانات كرسائل ICMP (ستناقش في الجزء 4-4-3).

لاحظ أن وحدة بيانات IP بها 20 بايتاً للترويسة (بافتراض عدم وجود خيارات). إذا كانت وحدة البيانات تحمل قطعة TCP فذلك يعني أن كل وحدة بيانات غير مجزأة تحمل ما مجموعه 40 بايتاً للترويسة (20 بايتاً لترويسة IP و20 بايتاً لترويسة TCP) بالإضافة إلى رسالة طبقة التطبيقات.

تجزئة وحدة بيانات IP

سنرى في الفصل الخامس أنه ليست كل بروتوكولات طبقة ربط البيانات يمكن أن تحمل رزم طبقة شبكة بنفس الحجم. يمكن أن تحمل بعض البروتوكولات رزم بيانات كبيرة، بينما يمكن أن تحمل بروتوكولات أخرى رزماً صغيرة فقط. على سبيل المثال يمكن أن تحمل إطارات إيثرنت في حدود 1500 بايت من بايتات البيانات، بينما لا يمكن أن تحمل إطارات بعض وصلات شبكة المنطقة الواسعة (Wide-Area Network (WAN)) أكثر من 576 بايتاً. الكمية القصوى للبيانات التي يمكن أن يحملها إطار طبقة ربط البيانات تدعى وحدة الإرسال القصوى (MTU). ولأن كل وحدة بيانات IP تكون مُغلّفة ضمن إطار طبقة ربط البيانات لنقلها من موجّه لآخر تمثل الكمية MTU ببروتوكول طبقة ربط البيانات حدّاً أقصى لطول وحدة بيانات IP. لا يُشكِّل وجود مثل هذا الحد الأقصى على حجم وحدة بيانات IP مشكلةً كبيرةً. وإنما تأتي المشكلة من استخدام بروتوكولات مختلفة في طبقة ربط البيانات على الوصلات المختلفة على طول المسار بين المصدر والوجهة وكلٌّ منها تستخدم قيماً مختلفة لـ MTU.

ولفهم قضية التمرير بطريقة أفضل تخيل بأنك موجّهٌ يربط عدّة وصلات لكلٍّ منها بروتوكول مختلف لطبقة ربط البيانات ولها حجم أقصى مختلف لوحدة النقل (MTU). افترض أنك عندما تستلم وحدات بيانات IP من وصلة واحدة تقوم بفحص جدول التمرير لديك لتقرير وصلة المخرج. افترض أن وصلة المخرج هذه لها MTU أصغر من طول وحدة بيانات IP. كيف لك أن تضغط وحدة بيانات IP تلك

الكبيرة جداً في حقل الحمل الآجر لإطار طبقة ربط البيانات؟ يتمثل الحل في تجزئة وحدة بيانات IP الأصلية إلى وحدتين أو أكثر تكون أصغر حجماً ، ثم تغليف كلٍّ منهما في إطار منفصل يُرسَل على وصلة المخرج. يُطلَق على كل وحدة من وحدات البيانات الأصغر تلك جزءاً (fragment).

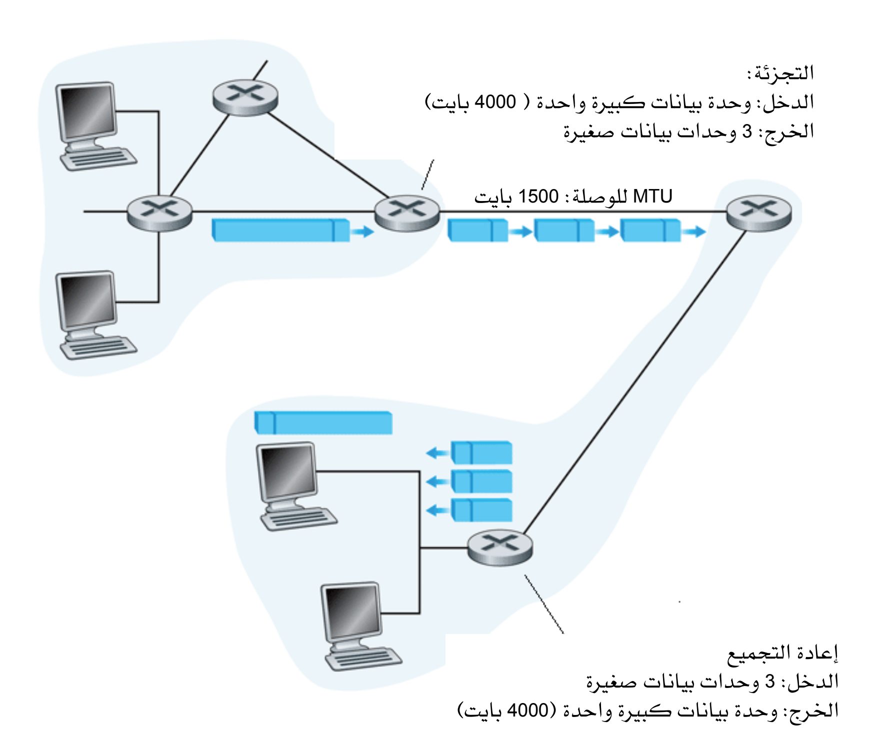
يجب إعادة تجميع الأجزاء قبل تسليمها إلى طبقة النقل في الوجهة. وفي واقع الأمر يتوقع كلٌّ من TCP وUDP استلام قطعاً كاملة غير مجزأة من طبقة الشبكة. لقد أحس مصممو بروتوكول IPv4 أن إعادة تجميع الوحدات الجزئية في الموجّهات سيؤدي إلى تعقيد ملحوظ في البروتوكول مما يقلل كفاءته. تخيل نفسك مكان الموجّه، هل تريد القيام بإعادة تجميع الوحدات الجزئية بجانب كل شيء آخر يفترض أن تقوم به؟ بالتمسّك بمبدأ إبقاء الشبكة الرئيسة بسيطة، قرّر مصممو IPv4 إيكال مهمة إعادة تجميع الوحدات الجزئية إلى الأنظمة الطرفية بدلاً من موجّهات الشبكة.

عندما يستلم مضيف الوجهة سلسلة من وحدات البيانات من نفس المصدر يحتاج لتحديد ما إذا كانت أيٌّ من تلك الوحدات هي أجزاء من وحدة أصلية أكبر. إذا كانت بعض وحدات البيانات الواصلة أجزاء، فعليه أيضاً أن يقرّر متى استلم الجزء الأخير وكيف يجب أن توضع الأجزاء التي استلمها سويةً لتشكيل وحدة البيانات الأصلية. وللسماح لمضيف الوجهة بأداء هذه المهمّة، وضع مصممو بروتوكول الإنترنت (النسخة 4) حقول المعرّف (identifier) والعلم (flag) والعنوان النسبي للتجزئة (fragmentation offset) في ترويسة وحدة بيانات بروتوكول الإنترنت. عند تكوين وحدة بيانات يختم مضيف الإرسال وحدة البيانات بعدد تعريفي (المعرّف) بالإضافة إلى عناوين المصدر والوجهة. ويزيد مضيف الإرسال العدد التعريفي لكل وحدة بيانات يرسلها بعد ذلك. وعندما يحتاج موجّه لتجزئة وحدة بيانات تختم كل الوحدات الجزئية الناتجة بنفس عنوان المصدر وعنوان الوجهة والعدد التعريفي لوحدة البيانات الأصلية. بعد أن تستلم الوجهة سلسلة من وحدات البيانات من نفس مضيف الإرسال، تفحص العدد التعريفي لكل وحدة بيانات لتقرير أيٍّ منها يمثل في الحقيقة جزءاً من وحدة أكبر. ولأن بروتوكول الإنترنت

يوفر خدمة غير موثوق فيها لنقل البيانات، فيمكن ألا يصل جزء أو أكثر من تلك الأجزاء إلى الوجهة. لهذا السبب، وحتى يتمكّن مضيف الوجهة من التأكد من استلام الجزء الأخير للوحدة الأصلية ، توضع قيمة حقل العلم 0 بهذا الجزء بينما تكون قيمة هذا الحقل في كل الأجزاء الأخرى 1. أيضاً لكي يتمكّن مضيف الوجهة من تقرير ما إذا كان جزء قد فقد (وأيضاً من إعادة تجميع الأجزاء في ترتيبها الصحيح)، يُستعمَل حقل العنوان النسبي ليحدّد أين يقع هذا الجزء ضمن وحدة البيانات الأصلية.

يوضح الشكل 4-14 مثالاً لذلك حيث تصل وحدة بيانات مكونة من 4000 بايت (20 بايتاً للترويسة بالإضافة إلى 3980 بايتاً حمل آجر) إلى موجّه، وعليه أن يرسلها إلى وصلة ذات حجم أقصى لوحدة البيانات MTU يساوي 1500 بايت. يشير هذا ضمناً إلى أن بايتات البيانات الـ 3980 في الوحدة الأصلية يجب أن تقسم إلى ثلاثة أجزاء منفصلة (كلٌّ منها سيمثل أيضاً وحدة بيانات IP). افترض أن الوحدة الأصلية مختومة بعدد تعريفي قيمته 777. يوضح الجدول 4-2 خصائص الأجزاء الثلاثة. تعكس القيم في الجدول 4-2 المطلب بأن كمية بيانات الحمل الآجر الأصلية في كل جزء فيما عدا الجزء الأخير يجب أن تكون مضاعفات لـ 8 بايتات، وأن تحدد قيم حقل العنوان النسبي بوحدات مكونة من 8 بايتات.

عند الوجهة تعبر بيانات الحمل الآجر لوحدة البيانات فقط إلى طبقة النقل بعد أن تكون طبقة الشبكة قد أعادت بناء الوحدة الأصلية بالكامل. إذا لم يصل جزء أو أكثر إلى الوجهة فسوف تسقط وحدة البيانات التي ينقصها ذلك الجزء بالكامل ولا تمرر إلى طبقة النقل. لكن – كما عرفنا في الفصل السابق – إذا استُخدم بروتوكول TCP في طبقة النقل فإنه سيعوّض هذا الفقد بجعل المصدر يعيد إرسال البيانات المفقودة من جديد.



الشكل 4-14 تجزئة وإعادة تجميع وحدة بيانات بروتوكول IP.

جزء الرزمة	عدد البايتات	الرقم التعريفي	قيمة العنوان النسبي	قيمة بت العَلَم
الأول	1480 بايت من بيانات رزمة IP	777	0 (أي يجب وضع البيانات في البداية عند البايت 0)	1 (أي أنه ليس الجزء الأخير في الرزمة)
الثاني	1480 بايت من البيانات	777	135 (أي يجب وضع البيانات عند البايت 1480، حيث أن 8 × 135 = 1480)	1 (أي أنه ليس الجزء الأخير في الرزمة)
الثالث	1020 بايت من البيانات (وهي البايتات المتبقية)	777	370 (أي يجب وضع البيانات عند البايت 2960، حيث أن 8 × 370 = 2960)	0 (أي أنه الجزء الأخير في الرزمة)

الجدول 4-2 الوحدات الجزئية الناتجة.

لقد عرفنا للتوِّ أن التجزئة في بروتوكول الإنترنت تلعب دوراً مهماً في توصيل العديد من التقنيات المتباينة لطبقة ربط البيانات. لكن التجزئة لها ثمنها أيضاً. فهي أولاً تُعقّد الأنظمة الطرفية والموجّهات، حيث يتعين أن تصمم بحيث يمكنها القيام بالتجزئة وإعادة تجميع وحدات البيانات. ثانياً يمكن أن تُستخدَم التجزئة لشن هجمات قاتلة لحجب الخدمة (DoS)، حيث يرسل المهاجم سلسلة من الوحدات الجزئية الغريبة وغير المتوقّعة. وكمثال تقليدي لذلك ما يعرف بهجوم Jolt2، حيث يرسل المهاجم فيضاً من الوحدات الجزئية الصغيرة - التي ليس لأيٍّ منها القيمة "صفر" في حقل العنوان النسبي - إلى مضيف الهدف. يمكن أن ينهار مضيف الهدف وهو يحاول إعادة بناء وحدات بيانات من تلك الوحدات الجزئية التالفة. وفي نوع آخر من تلك الحيل يتم إرسال وحدات جزئية متداخلة (أي لها قيم عناوين نسبية لا تسمح بإعادة وضع الوحدات الجزئية بشكلٍ صحيح). يمكن أن تنهار أنظمة التشغيل الضعيفة - أي التي لا تعرف ماذا تفعل مع تلك الوحدات الجزئية المتداخلة [Skoudis 2006]. وسنرى في نهاية هذا الجزء أن النسخة الجديدة من بروتوكول الإنترنت IPv6 تخلصت بالجملة من التجزئة، وذلك لتحسين معالجة وحدات البيانات وجعله أقل عرضة للهجمات.

يوجد على موقع الويب لهذا الكتاب (http://www.awl.com/kurose-ross) برنامج جافا صغير لتوليد وحدات جزئية. من خلال هذا البرنامج يحدد المستخدم حجم وحدة البيانات الواصلة وقيمة MTU وعدد تعريفي لتلك الرزمة، فيولّد البرنامج الوحدات الجزئية آلياً.

4-4-2 العنونة في بروتوكول IPv4

سنحوّل انتباهنا الآن إلى عناوين IPv4. بالرغم من أنك قد تعتقد بأن العنونة يجب أن تكون موضوعاً بسيطاً إلا أننا نأمل أن تقتنع مع نهاية هذا الفصل بأن هذا الموضوع ليس فقط مثيراً ودقيقاً بل وأنه يحظى بقدرٍ كبير من الأهمية في الإنترنت. من المعالجة الممتازة لموضوع عنونة IPv4 المقال [Com Addressing 20073] والفصل الأول في كتاب [Stewart 1999].

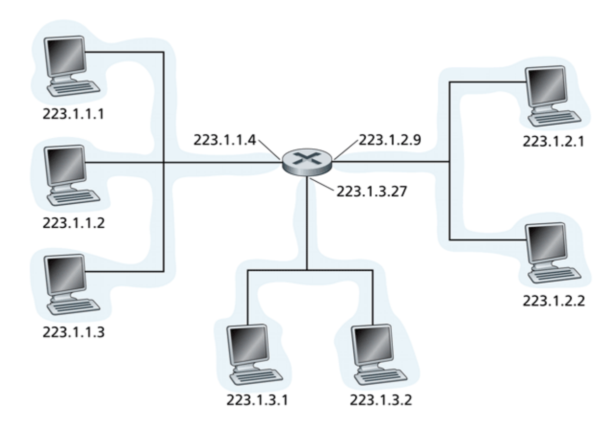
لكن قبل مناقشة عنونة IP سنحتاج لقول بضع كلمات حول كيفية توصيل المضيفات والموجّهات بالشبكة. يُوصّل المضيف بالشبكة عادةً بوصلة واحدة والتي عن طريقها يرسل بروتوكول IP على المضيف وحدات البيانات للشبكة. يطلق على نقطة تلاقي المضيف والوصلة المادية "واجهة" (interface). لننظر الآن إلى موجّه وواجهاته. نظراً لأن وظيفة الموجّه هي استقبال وحدة بيانات على وصلةٍ ما وإعادة إرسالها على وصلة أخرى، فإن الموجّه يوصّل بالشبكة بالضرورة عن طريق وصلتين أو أكثر. يطلق أيضاً على نقطة تلاقي الموجّه وأي وصلة مادية "واجهة"، وبالتالى يكون للموجّه واجهات متعددة (واحدة لكل وصلة). لأن كل مضيف وموجّه قادرٌ على إرسال واستلام وحدات بيانات IP ، يتطلب برتوكول IP أن يكون لكل واجهة لمضيف أو موجّه عنوان IP خاص بها (أي أنه من الناحية الفنية يرتبط عنوان IP بواجهة ما وليس بالمضيف أو بالموجّه الذي يحتوي على تلك الواجهة).

يتكون كل عنوان IP من 32 بتاً (أي أربعة بايتات)، ولهذا يكون العدد الإجمالي لعناوين IP المحتملة = 2^{32}. بتقريب 2^{10} إلى 10^{3} فمن السّهل ملاحظة أن هناك حوالي 4 بلايين عنوان IP محتمل. وفي العادة تكتب تلك العناوين في صيغة عشرية منقوطة، أي يكتب كل بايت من العنوان في شكلٍ عشري ويفصل بنقطة عن البايتات الأخرى في العنوان. على سبيل المثال يتكون العنوان 193.32.216.9 من أربعة بايتات: تمثل القيمة 193 (بنظام العد العشري) البايت الأول، والقيمة 32 البايت الثاني، وهكذا. والصيغة المكافئة لهذا العنوان بنظام العد الثنائي هي

11000001 00100000 11011000 00001001

يجب أن يكون لكل واجهة على كل مضيف وموجّه في شبكة الإنترنت العالمية عنوان IP فريدٌ عالمياً (أي غير مكرر). يُستثنى من ذلك الواجهات وراء أنظمة NAT كما سنناقش في نهاية هذا الجزء. لكن لا يمكن اختيار تلك العناوين بطريقة عشوائية. فجزء من عنوان بروتوكول الإنترنت للواجهة يحدد الشبكة الفرعية التي تتصل بها تلك الواجهة.

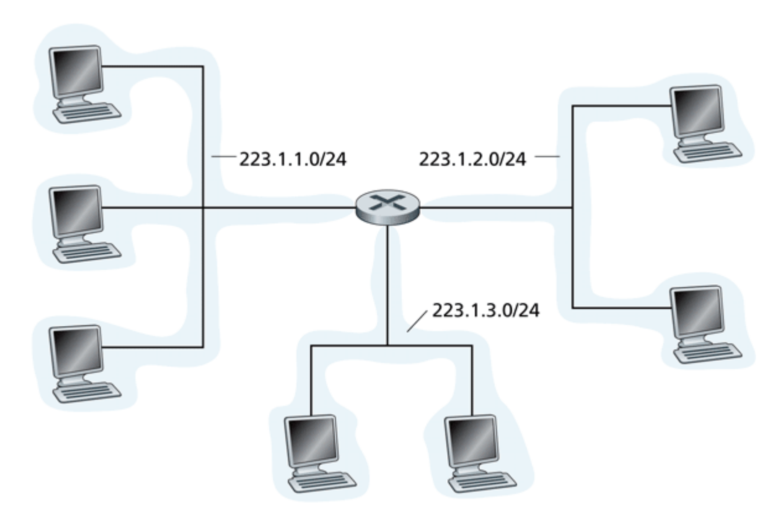
يبين الشكل 4-15 مثالاً لعنونة IP للواجهات. حيث يظهر موجّه واحد (بثلاث واجهات) يربط بين سبعة مضيفات. لنلقِ نظرة فاحصة على عناوين بروتوكول الإنترنت المخصّصة لواجهات المضيفات والموجّهات حيث توجد عدّة أمور يجب ملاحظتها. إن عنوان IP لكلٍّ من واجهات المضيفات الثلاثة في الجزء الأعلى يساراً في الشكل 4-15 وواجهة الموجّه التي يرتبط بها كلٌّ منهم له الصيغة 223.1.1.xxx.



الشكل 4-15 عناوين الواجهات والشبكات الفرعية.

أي أن لكلٍّ منها نفس الـ 24 بتاً من ناحية اليسار. ترتبط الوصلات الأربعة أيضاً ببعضها البعض من قِبَل شبكة لا تحتوي على موجّهات. يمكن أن تكون هذه الشبكة على سبيل المثال شبكة إيثرنت محلية حيث توصل الواجهات بمجمّع إيثرنت (hub) أو محول إيثرنت (switch) (انظر الفصل الخامس). في مصطلحات بروتوكول الإنترنت تعد الشبكة التي تصل بين واجهات المضيفات الثلاثة وأحد واجهات الموجّه شبكة فرعية [RFC 950] (يطلق أيضاً على تلك الشبكة شبكة IP أو ببساطة شبكة). تخصّص عنونة IP العنوان 223.1.1.0/24 لهذه الشبكة الفرعية، وأحياناً يطلق على الصيغة 24/ قناع الشبكة الفرعية (subnet mask)، وهي تشير إلى أن الـ 24 بتاً من يسار العنوان تمثل عنوان الشبكة الفرعية. تتكون الشبكة الفرعية 223.1.1.0/24 من ثلاث واجهات للمضيفات (223.1.1.1، 223.1.1.2، 223.1.1.3) وواجهة موجّه واحدة (223.1.1.4). ويجب أن يكون لأي مضيفات أخرى

توصَّل بالشبكة الفرعية 223.1.1.0/24 عنوان بالصيغة 223.1.1.xxx. توجد شبكتان فرعيتان إضافيتان في الشكل 4-15: شبكة 223.1.2.0/24 وشبكة 224.1.3.0/24 يوضح الشكل 4-16 شبكات IP الفرعية الثلاثة الموجودة في الشكل 4-15.



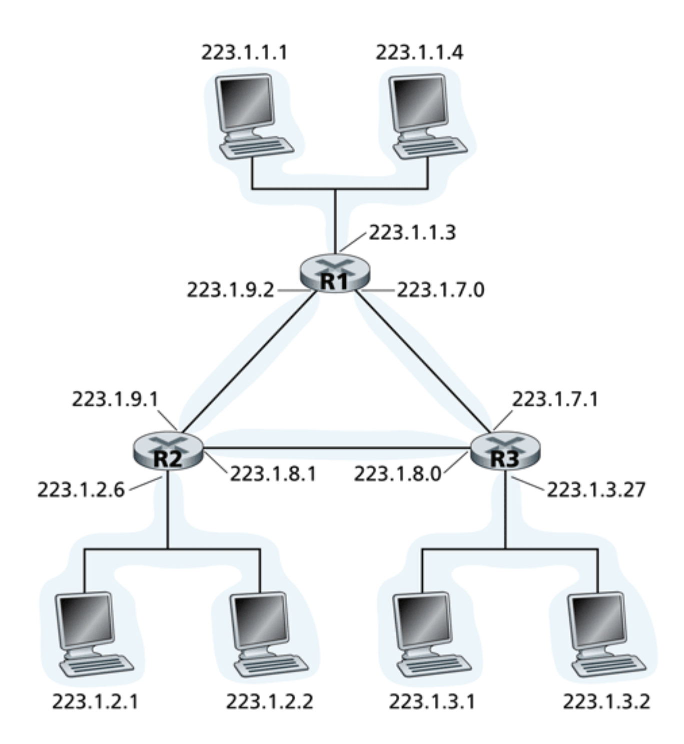
الشكل 4-16 عناوين الشبكات الفرعية.

لا يقتصر تعريف بروتوكول IP لشبكة فرعية على قطع الإيثرنت (Ethernet segments) التي توصل عدة مضيفات إلى واجهة موجّه. ولتوضيح ذلك انظر الشكل 4-17 حيث يوجد ثلاثة موجّهات متصلة مع بعضها البعض بوصلات من نوع "نقطة إلى نقطة". كل موجّه له ثلاث واجهات: واحدة لكل وصلة "نقطة إلى نقطة" وواحدة لوصلة الإذاعة التي توصل الموجّه مباشرة مع زوج من المضيفات. ما هي إذن الشبكات الفرعية الموجودة هنا؟ توجد ثلاث شبكات فرعية بالعناوين 223.1.1.0/24 و223.1.2.0/24 و223.1.3.0/24، وهي مشابهةٌ لتلك الموجودة في الشكل 4-15. كما أن هناك ثلاث شبكات فرعية إضافية في هذا المثال: شبكة فرعية 223.1.9.0/24 للواجهات التي توصّل الموجّه R1 مع R2 وشبكة فرعية أخرى 223.1.8.0/24 للواجهات التي توصل الموجّه R3 مع R2 وشبكة فرعية ثالثة 223.1.7.0/24 للواجهات التي توصل الموجّه R3 مع R1. في شبكة عامة مكونة من

مضيفات وموجّهات، يمكن استخدام الوصفة التالية لتعريف الشبكات الفرعية التي تتضمنها تلك الشبكة:

لتحديد الشبكات الفرعية نفصل كل واجهة من كل المضيفات والموجّهات، وبالتالى تصبح الشبكة عدداً من الجزر المعزولة، حيث تمثل الواجهات النهايات للنقاط الطرفية لتلك الجزر. ويطلق على كل جزيرة من تلك الجزر المعزولة شبكة فرعية.

إذا طبّقنا هذه القاعدة على الشبكة في الشكل 4-17 فسنحصل على ست جزر تمثل شبكات فرعية.



الشكل 4-17 ثلاثة موجّهات تربط بين ست شبكات فرعية.

من المناقشة السابقة يتضح أن شبكة منظمة (كشركة أو مؤسسة أكاديمية) مكونة من عددٍ من قطع الإيثرنت ووصلات "نقطة إلى نقطة" سيكون فيها شبكات فرعية متعدّدة، وسيكون للأجهزة على كل شبكة فرعية نفس عنوان الشبكة الفرعية. يمكن من حيث المبدأ أن تأخذ الشبكات الفرعية المختلفة عناوين شبكة فرعية مختلفة جداً. لكن عملياً ستشترك عناوين الشبكات الفرعية في أغلب الأحيان في أمور كثيرة. لفهم السبب دعنا نلفت الانتباه لكيفية معالجة العنونة في شبكة الإنترنت العالمية.

تعرف استراتيجية تخصيص عناوين الإنترنت بأسلوب التوجيه اللانوعي بين النطاقات (Classless InterDomain Routing (CIDR)) [RFC 4632]، وهي تعميم لفكرة عنونة الشبكة الفرعية. كما هو الحال مع عنونة الشبكة الفرعية يقسّم العنوان المكون من 32 بتاً إلى جزأين ويكتب أيضاً في الصيغة العشرية المنقوّطة *a.b.c.d/x* حيث تشير *x* إلى عدد البتات في الجزء الأول من العنوان.

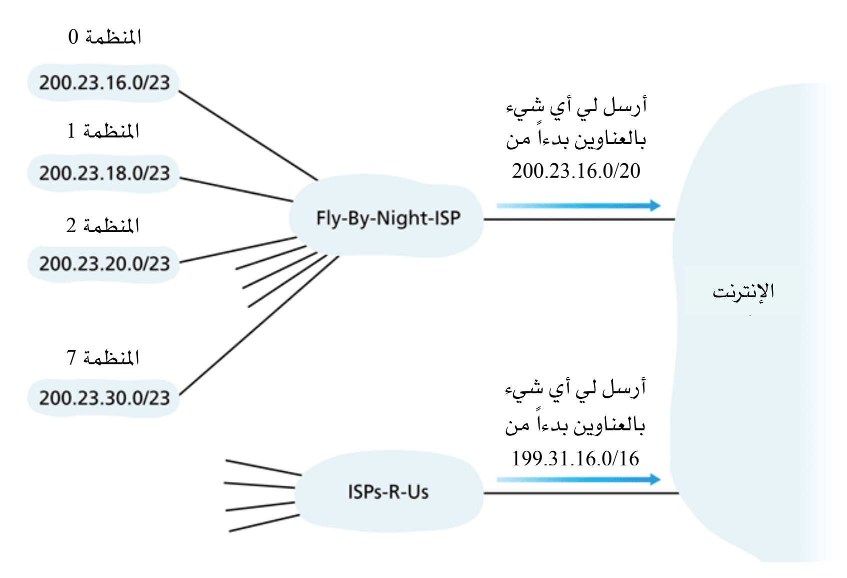
تمثل البتّات الأعلى رتبة، وعددها *x* بت، من عنوان ما بالصيغة *a.b.c.d/x* عنوان الشبكة التي ينتمي لها هذا العنوان، وفي أغلب الأحيان يشار إليها باسم البادئة (أو بادئة الشبكة) للعنوان. وعادة ما يخصّص لمنظمة ما عددٌ من العناوين المتجاورة – أي التي لها نفس البادئة (ويطلق عليها كتلة العناوين (address block)). في تلك الحالة ستشترك عناوين IP للأجهزة الموجودة ضمن شبكة المنظمة في البادئة. سنرى عندما نغطّي بروتوكول التوجيه BGP في الجزء 4-6 أن الموجّهات خارج شبكة المنظمة تفحص فقط بتات البادئة *x* هذه عند اختيار مسار وحدات البيانات. وهذا يُخفض حجم جداول التوجيه إلى حدٍّ كبير في تلك الموجّهات، لأنه سيكفي مُدخل واحد فقط بالصيغة *a.b.c.d/x* بالجداول لإرسال وحدات البيانات لأي وجهة ضمن المنظمة.

المبادئ في الواقع العملي (Principles in Practice)

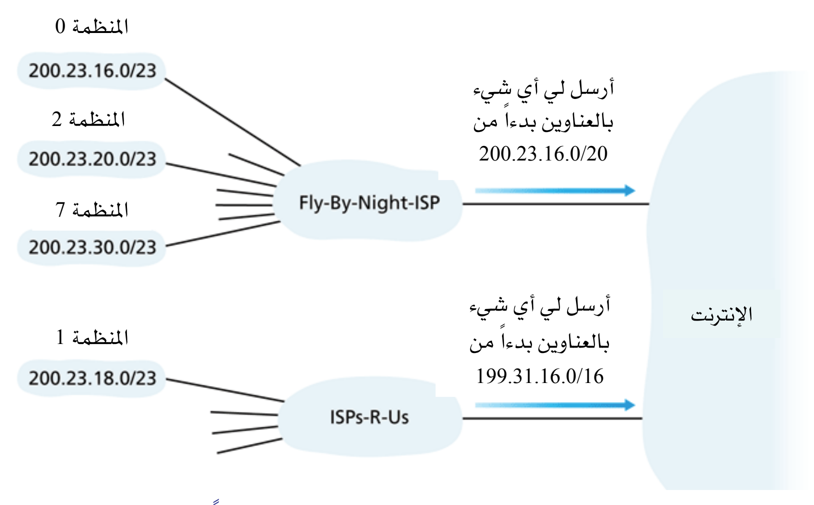
هذا المثال لموفر لخدمة الإنترنت يوصِّل ثماني منظمات إلى الإنترنت ويوضح بشكلٍ رائع كيف خُصّصت عناوين CIDR بعناية لتسهيل التوجيه. افترض - كما هو موضح في الشكل 4-18 أن موفر خدمة الإنترنت (والذي سنطلق عليه Fly-By-Night-ISP) يعلن للعالم الخارجي بأنه يجب أن ترسل إليه أي رزم الـ 20 بتاً الأولى في عناوينها تطابق 200.23.16.0/20. لا يلزم أن يعرف بقيّة العالم أن ضمن كتلة العناوين 200.23.16.0/20 توجد في الحقيقة ثماني منظمات أخرى لكلٍّ منها شبكاتها الفرعية الخاصة بها. غالباً ما يُشار إلى القدرة على استعمال بادئة واحدة بتجميع العناوين (address aggregation) (أيضاً تسمى تجميع أو دمج المسارات).

يعمل هذا الأسلوب بطريقة جيدة للغاية عندما تُخصّص العناوين على شكل كتل لموفري خدمة الإنترنت ومنهم إلى المنظمات الزبائن. لكن ماذا يحدث عندما لا تُخصص العناوين بهذا الأسلوب الهرمي (hierarchical)؟

ماذا كان سيحدث – على سبيل المثال – إذا امتلك موفر خدمة الإنترنت Fly-By-Night-ISP موفر خدمة إنترنت آخر يدعى ISPs-R-Us ثم جعل المنظمة 1 تتصل بالإنترنت من خلال ذلك التابع ISPs-R-Us؟ كما هو موضح في الشكل 4-18 يمتلك موفر خدمة الإنترنت التابع ISPs-R-Us كتلة العناوين؛ لكن ولسوء الحظ عناوين المنظمة 1 خارج هذه الكتلة. ما الذي يجب فعله هنا؟ بالتأكيد يمكن أن تعيد المنظمة 1 ترقيم كل موجّهاتها ومضيفاتها لتكون عناوينها ضمن كتلة موفر خدمة الإنترنت ISPs-R-Us. إلا أن هذا الحل مكلِّف، خصوصاً أن المنظمة 1 قد يعاد توصيلها في المستقبل عن طريق موفر خدمة إنترنت تابع آخر. إن الحل الذي غالباً ما يشتخدَم هو جعل المنظمة 1 تحتفظ بالعناوين في الكتلة 200.23.18.0/23. في هذه الحالة – كما هو مبين في الشكل 4-19 – يواصل موفر خدمة الإنترنت Fly-By-Night-ISP إعلان كتلة العناوين 200.23.16.0/20 ويواصل موفر خدمة الإنترنت ISPs-R-Us إعلان 199.31.16.0/16. ومن ناحية أخرى يعلن موفر خدمة الإنترنت ISPs-R-Us كتلة العناوين للمنظمة 1 أي 200.23.18.0/23. عندما ترى الموجّهات الأخرى في الإنترنت كتلة العناوين 200.23.16.0/20 (من Fly-By-Night-ISP) وكتلة العناوين 200.23.18.0/23 (من ISPs-R-Us) وتريد توجيه رزم لوجهة تقع في نطاق الكتلة 200.23.18.0/23، ستستعمل تطابق البادئة الأطول (longest prefix matching) (راجع الجزء 4-2-2) وتوجه نحو موفر خدمة الإنترنت ISPs-R-Us لأنه في هذه الحالة يمثل البادئة الأطول (أي الأكثر تحديداً) التي تطابق عنوان الوجهة.



الشكل 4-18 العنونة الهرمية وتجميع المسارات.



الشكل 4-19 ISPs-R-Us له مسار أكثر تحديداً للمنظمة 1.

يمكن اعتبار بقية البتات في العنوان على أنها تُميّز بين الأجهزة ضمن المنظمة التي لها نفس بادئة الشبكة. هذه البتات هي التي ستُفحص عند توجيه الرزم داخل المنظمة. وقد يكون (أو لا يكون) لهذه البتات ذات الرتبة الأدنى تركيب لتفريع شبكي إضافي كالذي ناقشناه من قبل. على سبيل المثال افترض أن البتات الـ 21 الأولى من العنوان a.b.c.d/21 تحدّد بادئة شبكة المنظمة وهي ثابتة في عناوين كل الأجهزة في تلك المنظمة. أما البتات الباقية الإحدى عشرة الأخرى فهي لتمييز المضيفات في المنظمة. قد يكون التركيب الداخلي لشبكة المنظمة بحيث تستخدم تلك البتات الإحدى عشرة في أقصى اليمين لعناوين الشبكات الفرعية ضمن المنظمة كما ذكرنا سابقاً. على سبيل المثال قد يشير العنوان a.b.c.d/24 إلى شبكة فرعية معينة ضمن المنظمة.

قبل استخدام أسلوب CIDR للعنونة كان جزء العنوان الذي يدل على الشبكة مقيداً بواحد من الأطوال 8 أو 16 أو 24 بتاً، وهو ما عرف بالعنونة النوعية (classful) وتعرف الشبكات التي تنتمي لكل نوع من هذه العناوين بالفئة A أو B أو C على الترتيب. لكن المطلب أن يكون طول الجزء الدال على الشبكة لعنوان محصوراً في تلك القيم (أي 1 أو 2 أو 3 بايتات) سبّب مشكلةً لدعم العدد المتزايد بسرعة من المنظمات التي تمتلك شبكات فرعية صغيرة ومتوسطة الحجم. كما أن تخصيص عناوين من الفئة C (/24) يدعم فقط $2^8 - 2 = 254$ مضيف كحد أقصى (حيث إن اثنين من تلك العناوين محجوزان للاستعمال الخاص) والذي قد يكون صغيراً جداً بالنسبة للعديد من المنظمات. في حين أن أقصى عدد تدعمه الفئة B (/16) من المضيفات يساوي 65634 مضيفاً والذي قد يعتبر كبيراً جداً لتلك المنظمات. بالتالي إذا استخدمنا هذا الأسلوب للعنونة فإن منظمة لديها فقط 2000 مضيفٍ ستحتاج إلى عنوان من الفئة B؛ الأمر الذي يؤدي لاستنزاف سريع لفضاء عناوين الفئة B واستخدام سيّئ للعناوين المخصّصة. على سبيل المثال ستستخدم المنظمة السابقة التي لديها 2000 مضيفٍ فقط 2000 عنوانٍ من عناوين الفئة B التي خصصت لها تاركة بذلك أكثر من 63000 عنوانٍ لا يمكن استخدامها من قِبَل منظمات أخرى.

سنكون مقصرين إذا لم نذكر نوعاً آخر من عناوين بروتوكول الإنترنت وهو عنوان الإذاعة 255.255.255.255. فعندما يرسل مضيف وحدة بيانات لهذا العنوان كعنوان الوجهة تُسلّم الرسالة إلى كل المضيفات على نفس الشبكة الفرعية، ويمكن أن ترسل الموجّهات الرسالة إلى الشبكات الفرعية المجاورة أيضاً (غير أن هذا الاختيار لا يُستخدَم عادة).

بعد أن درسنا عنونة بروتوكول الإنترنت بالتفصيل نحتاج لمعرفة كيفية حصول المضيفات والشبكات الفرعية على عناوينها في البداية. دعنا نبدأ بالنظر إلى كيفية حصول منظمة ما على كتلة عناوين لأجهزتها، ثم إلى كيفية حصول جهاز (كمضيف مثلاً) على عنوان من بين كتلة عناوين المنظمة.

الحصول على كتلة العناوين

لكي تحصل منظمة على كتلة عناوين للاستعمال ضمن شبكتها قد يتّصل المشرف على الشبكة أولاً بموفر خدمة الإنترنت لتخصيص عناوين من كتلة أكبر من العناوين التي تم تخصيصها لموفر الخدمة من قبل. على سبيل المثال افترض أن موفر خدمة إنترنت قد حصل على الكتلة 200.23.16.0/20. يقوم موفر خدمة الإنترنت بدوره بتقسيم تلك الكتلة إلى ثماني كتل متجاورة بأحجام متساوية ويعطي لكل منظمة من المنظمات الثماني التي يدعمها كتلة منها كما هو مبين أدناه (وللتوضيح قمنا بوضع خط تحت جزء الشبكة الفرعية لهذه العناوين):

كتلة موفّر خدمة الإنترنت	200.23.16.0/20	<u>11001000 00010111 0001</u>0000 00000000
المنظمة 0	200.23.16.0/23	<u>11001000 00010111 0001000</u>0 00000000
المنظمة 1	200.23.18.0/23	<u>11001000 00010111 0001001</u>0 00000000
المنظمة 2	200.23.20.0/23	<u>11001000 00010111 0001010</u>0 00000000
.....	.....	.....
المنظمة 7	200.23.30.0/23	<u>11001000 00010111 0001111</u>0 00000000

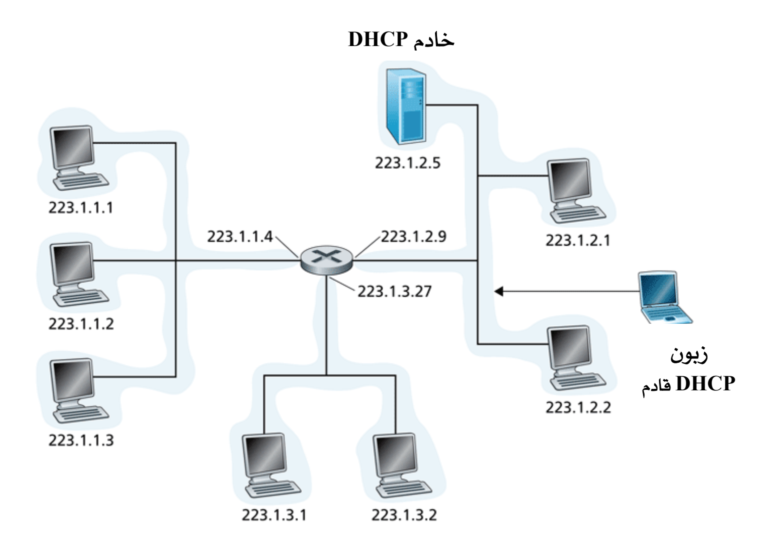
ليست هذه هي الطريقة الوحيدة للحصول على العناوين ولكنها إحدى الطرق. واضح أنه يجب أيضاً أن تكون هناك طريقة لموفر خدمة الإنترنت نفسه للحصول على كتلة عناوين، فهل هناك سلطة عالمية لها مسؤولية نهائية لإدارة فضاء عناوين الإنترنت وتخصيص كتل منها لموفري خدمة الإنترنت والمنظمات الأخرى؟ في الحقيقة نعم هناك سلطة! فعناوين الإنترنت مدارة تحت سلطة شركة الإنترنت للأسماء والأعداد المخصّصة (ICANN) [ICANN 2007] بناءً على الإرشادات الموجودة بـ RFC 2050. ودور هذه المنظمة اللاربحية ليس فقط تخصيص عناوين بروتوكول الإنترنت ولكن أيضاً إدارة خادمات أسماء النطاقات الجذرية (DNS root servers). كذلك تعمل على تخصيص أسماء النطاقات وحل النزاعات المتعلقة بها. تخصّص ICANN عناوين مكاتب تسجيل الإنترنت الإقليمية (مثل: ARIN، RIPE، APNIC، LACNIC) والتي تُشكِّل سويةً المنظمة المساندة للعناوين لـ (ICANN) [ASO-ICANN 2007] وتعالج تخصيص وإدارة العناوين ضمن مناطقها.

الحصول على عنوان مضيف: بروتوكول تهيئة المضيف الديناميكي (DHCP)

بعد أن تحصل منظمة على كتلة عناوين يمكنها تخصّيص عناوين لواجهات الموجّهات والمضيفات لديها. عادة ما يقوم المسؤول عن الشبكة بتهيئة عناوين IP للموجّهات يدوياً (غالباً ما يتم ذلك عن بُعد باستخدام أداة إدارة الشبكة). وبالمثل يمكن أيضاً تهيئة عناوين المضيفات بطريقة يدوية إلا أنه في أغلب الأحيان تستخدم هذه العملية بروتوكول DHCP لتهيئة المضيفات ديناميكياً [RFC 2131]. يسمح بروتوكول DHCP لمضيف بالحصول على عنوان IP آلياً. يمكن أن يهيئ المشرف على الشبكة بروتوكول DHCP بحيث يعطي دائماً نفس العنوان لمضيف ما في كل مرة يتصل بالشبكة، أو قد يخصّص العنوان مؤقتاً للمضيف وبالتالي سيكون العنوان مختلفاً في كل مرة. بالإضافة إلى مهمة تخصيص عناوين المضيفات يسمح بروتوكول DHCP أيضاً للمضيف بالحصول على معلومات إضافية مثل قناع شبكته الفرعية وعنوان الموجّه الأول (والذي يطلق عليه في أغلب الأحيان البوابة الاعتيادية (default gateway) وعنوان خادم أسماء النطاقات المحلي.

بسبب قدرة بروتوكول DHCP على أتمتة خصائص الشبكة فيما يتعلق بتوصيل مضيف بها، فإنه غالبا ما يطلق عليه بروتوكول "وصّل وشغّل". هذه القدرة تجعله جذّاباً جداً لمشرف الشبكة الذي بدونه كان سيؤدي تلك المهمة يدوياً! كما يتمتّع بروتوكول DHCP بالاستعمال الواسع الانتشار في شبكات الإتصال بالإنترنت السكني وفي الشبكات المحلية اللاسلكية، حيث تتصل المضيفات بالشبكة وتغادرها كثيراً. تصوّر على سبيل المثال الطالب الذي يحمل حاسباً نقالاً من غرفة مسكنه إلى المكتبة إلى قاعة الدروس. من المحتمل أنه في كل موقع سيوصل بشبكة فرعية جديدة ولذلك سيحتاج عنوان IP جديد في كل موقع. يناسب DHCP بطريقة مثالية تلك الحالة، حيث العديد من مستخدمي الشبكة يجيؤون ويغادرون، والتي تكون فيها العناوين مطلوبة لفترة محدودة فقط. بالمثل يفيد DHCP بنفس الطريقة في شبكات الوصول السكني لموفر خدمة إنترنت. تصور مثلاً موفر خدمة إنترنت سكني لديه 2000 عميلٍ لكن لا يتصل أكثر من 400 عميلٍ منهم بالإنترنت في نفس الوقت. في هذه الحالة بدلاً من الحاجة إلى كتلة من 2048 عنوانٍ يمكن لموفر الخدمة استخدام خادم DHCP الذي يخصّص العناوين ديناميكياً، وفي هذه الحالة سيحتاج فقط كتلة من 512 عنواناً (على سبيل المثال كتلة عناوين بالصيغة a.b.c.d/23). وسيقوم خادم DHCP بتحديث قائمة العناوين المتوفرة لديه بينما تلتحق المضيفات بالشبكة أو تغادرها. في كل مرة ينضمّ مضيف للشبكة يخصّص خادم DHCP عنواناً اعتباطياً من القائمة الحالية للعناوين المتوفرة، وفي كل مرة يغادر مضيّف الشبكة يرجع عنوانه إلى قائمة العناوين المتوفرة.

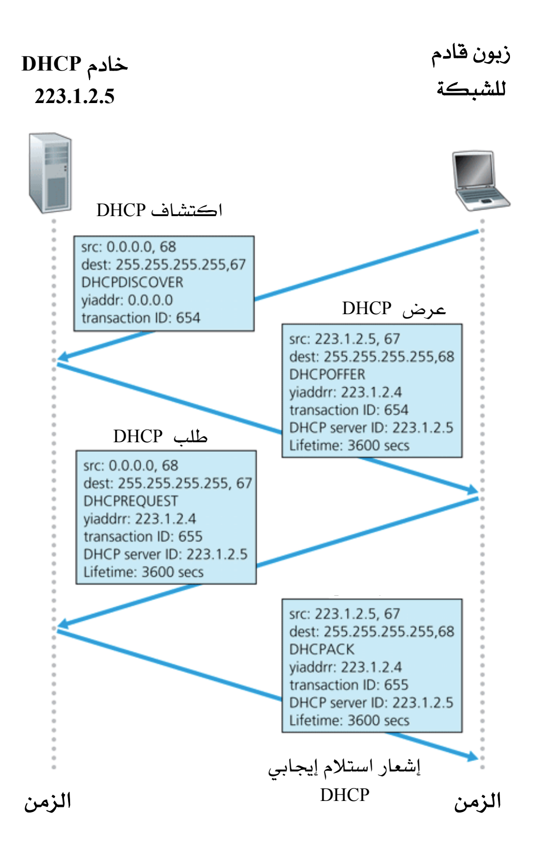
يستخدم بروتوكول DHCP بنية خادم/زبون. عادة ما يحتاج مضيف قادم للشبكة للحصول على معلومات تهيئة بما في ذلك عنوان IP له. في الحالة الأبسط يوجد في كل شبكة (بطريقة العنونة الموضحة في الشكل 4-17) خادم DHCP. أما إذا لم يوجد خادم DHCP فسنحتاج إلى وجود وكيل ترحيل (relay agent) يعرّف عنوان خادم DHCP. يوضح الشكل 4-20 خادم DHCP متصل بالشبكة الفرعية 223.1.2/24، ويعمل الموجّه وكيل ترحيل للزبائن الجديدة التي تصل للشبكات الفرعية 223.1.1/24 و223.1.3/24. سنفترض في مناقشتنا التالية أن خادم DHCP متوفر على الشبكة الفرعية.



الشكل 4-20 سيناريو التفاعل بين خادم وزبون DHCP.

لمضيف جديد قادم يعتبر بروتوكول DHCP عملية مكونة من أربع خطوات؛ كما يبين الشكل 4-21 لإعدادات الشبكة المعروضة في الشكل 4-20. في هذا الشكل (كما في "عنوان الإنترنت" الخاص بك) يشير الرمز yiaddr إلى العنوان المخصّص للمضيف الجديد القادم. والخطوات الأربعة هي:

- اكتشاف خادم DHCP: إن المهمّة الأولى للمضيف القادم حديثاً للشبكة هي أن يجد خادم DHCP الذي سيتفاعل معه. ويتم ذلك باستعمال رسالة اكتشاف DHCP والتي يرسلها المضيف ضمن رزمة UDP إلى المنفذ رقم 67. تغلّف رزمة UDP في رزمة بيانات IP. لكن إلى من يجب أن ترسل هذه الرزمة؟ إن المضيف لا يعرف على الإطلاق حتى عنوان IP للشبكة التي يتصل بها وبالأحرى لا يعرف عنوان خادم DHCP لهذه الشبكة. ولذا ينشئ زبون DHCP وحدة بيانات IP تحتوي على رسالته لاكتشاف DHCP سويّة مع عنوان IP الإذاعي للوجهة 255.255.255.255 وعنوان IP "لهذا المضيّف" (أي 0.0.0.0) للمصدر. يمرر زبون DHCP وحدة بيانات IP إلى طبقة ربط البيانات والتي تقوم بدورها بإذاعة الإطار الناتج إلى كل العقد المتصلة بالشبكة الفرعية (سنغطّي تفاصيل إذاعة طبقة ربط البيانات في الجزء 4-5).



الشكل 4-21 التفاعل بين خادم وزبون DHCP.

- عروض خادمات DHCP : يرد خادم DHCP الذي استلم رسالة اكتشاف DHCP على الزبون برسالة عرض DHCP تذاع إلى كل العقد على الشبكة الفرعية (مستعملاً العنوان 255.255.255.255 مرة أخرى للوجهة). (قد تحتاج لأن تفكر في سبب ضرورة إذاعة الرد من الخادم!). ونظراً لاحتمال وجود عدّة خادمات DHCP على الشبكة الفرعية قد يجد الزبون نفسه في وضع يحسد عليه حيث يستطيع الاختيار من بين عدّة عروض. تحتوي كل رسالة عرض من الخادم على الرقم التعريفي لرسالة الاكتشاف التي تلقاها، وعنوان بروتوكول الإنترنت المقترح للزبون، وقناع الشبكة، ومدة إيجار عنوان IP (أي المدة التي سيكون العنوان فيها صحيحاً – أي محجوزاً

لاستخدام المضيف ولا يمكن تخصيصه لمضيف آخر). من الشائع أن يضع الخادم مدة الإيجار عدّة ساعات أو أيام [Droms 1999].

- طلب DHCP: بعد أن يختار المضيف الواصل حديثاً للشبكة واحداً من عروض DHCP المقدمة له سيردّ عليها برسالة طلب DHCP ويضع بها نفس قيم بارامترات التهيئة الموجودة في العرض المختار.
- إشعار استلام DHCP: يردّ الخادم على رسالة طلب DHCP برسالة إشعار استلام DHCP مؤكّداً قيم البارامترات المطلوبة.

بمجرد استلام الزبون إشعار استلام DHCP يكون التفاعل بين الزبون والخادم قد اكتمل، ويمكن أن يستعمل الزبون عنوان IP المخصص له من خادم DHCP حتي تنتهي مدة الإيجار. ولأن الزبون قد يرغب في استعمال عنوانه بعد انتهاء مدة الإيجار يوفر DHCP أيضاً آلية تسمح للزبون بتجديد إيجار عنوان IP.

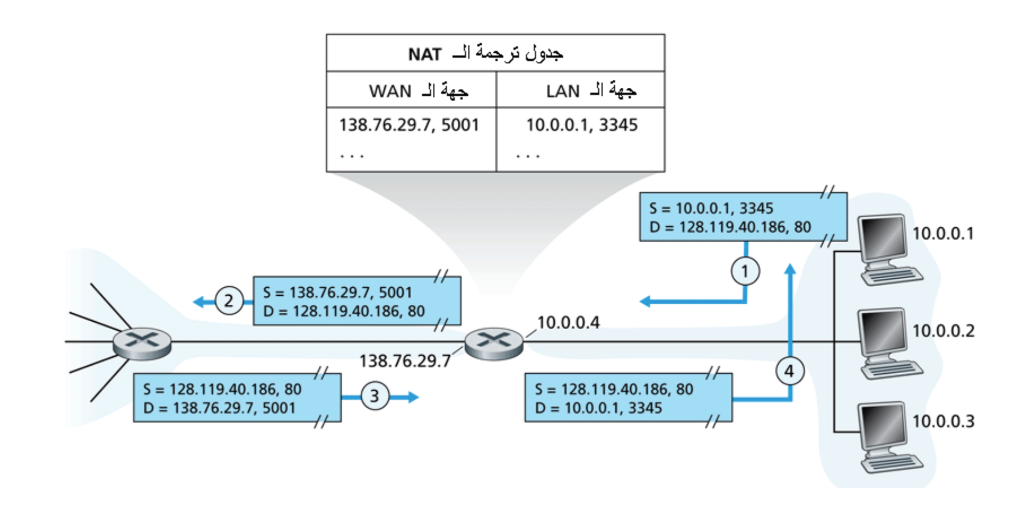
تتضح الفائدة الجليلة لخاصية "وصل وشغل" في بروتوكول DHCP إذا ما أخذنا في الاعتبار أن البديل هو تهيئة عنوان IP للمضيّف يدوياً. مثلاً تصور طالباً ينتقل من قاعة الدروس إلى المكتبة إلى غرفة المسكن مع حاسب نقال وفي كل موقع يتصل بشبكة فرعية جديدة فإنه يحتاج إلى عنوان IP جديد في كل موقع. من المستحيل تصور أن مدير النظام يجب أن يعيد تهيئة الحاسبات النقالة في كل موقع، كما أن الكثير من الطلاب (ما عدا أولئك الذين يأخذون مادة شبكات الحاسب!) ليس لديهم خبرة لتهيئة حاسباتهم النقالة يدوياً. ومع ذلك يعاني بروتوكول DHCP من بعض أوجه القصور من منظور قابلية الحركة. فمثلاً لا يمكن الاحتفاظ بتوصيلة TCP لعقدة تتحرّك بين شبكات فرعية لأنها تحصل على عنوان IP جديد من DHCP في كل مرة توصَّل بشبكة فرعية جديدة. في الفصل السادس سنفحص بروتوكول IP النقّال كامتداد حديث للبنية التحتية لبروتوكول IP يسمح لعقدة متنقلة باستعمال عنوان دائم وحيد بينما تتحرّك بين الشبكات الفرعية. يمكنك الاطلاع على التفاصيل الإضافية حول بروتوكول DHCP في [Droms 1999] و[dhc 2007]. يوجد برنامج مصدر مفتوح (open source code)

لتحقيق بروتوكول DHCP من اتحاد نظم الإنترنت (Internet Systems Consortium) [ISC 2007].

ترجمة عناوين الشبكة (NAT)

بعد أن ناقشنا عناوين الإنترنت وصيغة وحدة بيانات IPv4 ندرك الآن جيداً أن كل جهاز يعمل ببروتوكول IP يحتاج إلى عنوان IP. ومع انتشار شبكات سوهو SOHO (المكتب الصغير والمكتب المنزلي (small office home office) فإن ذلك يشير ضمناً إلى أنه حينما تريد سوهو تركيب شبكة اتصالات محلية لتوصيل عدد من الأجهزة فمن الضّروري أن يخصّص موفّر خدمة الإنترنت مدى من العناوين لتغطية كل أجهزة تلك الشبكة. لو اتسعت تلك الشبكة الفرعية أكثر (على سبيل المثال إذا اشترى الأطفال في المنزل بالإضافة إلى حاسباتهم الخاصة أجهزة PDA وهواتف IP ولعبة Game Boys للشبكة) فإنها ستحتاج إلى تخصيص مجموعة أكبر من العناوين. لكن ماذا لو أن موفر خدمة الإنترنت قد خصّص الأجزاء المتاخمة لعناوين شبكة سوهو الحالية لشبكة أخرى؟ وماذا يحتاج صاحب المنزل لمعرفة كيف يدير عناوين IP في المقام الأول؟ لحسن الحظ هناك طريقة بسيطة لتخصيص العناوين والتي لاقت استعمالاً واسعاً جداً في مثل هذه السيناريوهات يطلق عليها ترجمة عناوين الشبكة (NAT) [RFC 2663; RFC 3022].

يوضح الشكل 4-22 كيفية عمل موجّه مزوّد بـ NAT. توجد واجهة لموجّه الـ NAT القابع في المنزل كجزء من شبكة المنزل على الجانب الأيمن للشكل 4-22. تتم العنونة داخل شبكة المنزل بالضبط كما رأينا من قبل؛ فالواجهات الأربعة في شبكة المنزل لها نفس عنوان الشبكة الفرعية 10.0.0/24. يمثل فضاء العناوين 10.0.0.0/8 أحد ثلاثة أجزاء لعناوين IP المحجوزة في [RFC 1918] للشبكات الخاصّة أو لمنطقة بعناوين خاصّة كشبكة المنزل في الشكل 4-22. والمقصود بـ"منطقة بعناوين خاصّة" هنا شبكة يكون لعناوينها معنى فقط لدى الأجهزة الموجودة ضمن تلك الشبكة. كي نعي أهمية ذلك، تذكّر أن هناك مئات الآلاف



الشكل 4-22 ترجمة عنوان الشبكة.

من الشبكات المنزلية التي يستعمل الكثير منها نفس فضاء العناوين 10.0.0.0/24. يمكن أن ترسل الأجهزة الموجودة بشبكة منزلية لبعضها البعض باستعمال العنونة 10.0.0.0/24، ولكن واضح أنه لا يمكن أن تستعمل الرزم المرسلة خارج نطاق الشبكة المنزلية إلى شبكة الإنترنت العالمية الأكبر هذه العناوين (لا كمصدر ولا كوجهة) نظراً لأن هناك مئات الآلاف من الشبكات تستخدم نفس تلك الكتلة من العناوين. أي أن العناوين 10.0.0.0/24 يمكن أن يكون لها معنى فقط في نطاق الشبكة المنزلية المحددة. لكن إذا كانت العناوين الخاصّة لها معنى فقط ضمن الشبكة المحددة فكيف تعالج العنونة عند إرسال أو استلام رزم من الإنترنت العالمية (حيث يكون من الضروري استعمال عناوين فريدة)؟ يكمن الجواب في فهم نظام ترجمة عناوين الشبكة NAT.

لا يبدو الموجّه القادر على ترجمة عناوين الشبكة NAT للعالم الخارجي كموجّه، وإنما يتصرّف كجهاز واحد له عنوان IP وحيد. في الشكل 4-22، كل البيانات التي تغادر موجّه الشبكة الأم إلى الإنترنت الأكبر تستخدم العنوان 138.76.29.7 كعنوان المصدر، كما أن البيانات القادمة من الإنترنت إلى الموجّه لها العنوان 138.76.29.7 كعنوان الوجهة. بشكلٍ أساسي يحجب موجّه الـ NAT

تفاصيل الشبكة المنزلية عن العالم الخارجي. (قد تتساءل ومن أين تحصل حاسبات الشبكة المنزلية على عناوينها؟ ومن أين يحصل الموجّه على عنوانه الوحيد؟ في أغلب الأحيان يكون الجواب هو نفسه "عن طريق DHCP"! يحصل الموجّه على عنوانه من خادم DHCP لمزوّد خدمة الإنترنت، ويشغل الموجّه خادم DHCP لتزويد العناوين إلى الحاسبات ضمن فضاء عناوين الشبكة المنزلية التي تقع في نطاق تحكم موجّه DHCP).

إذا كانت كل وحدات البيانات التي تصل إلى موجّه NAT من الشبكة الواسعة النطاق (WAN) لها نفس عنوان IP للوجهة (بالتحديد عنوان واجهة موجّه NAT التي تتصل بالشبكة الواسعة النطاق) فكيف يعرف الموجّه المضيف الداخلي الذي يجب أن يرسل له وحدة البيانات؟ تكمن الحيلة هنا في استعمال جدول ترجمة NAT في الموجّه يتضمن في مدخلاته أرقام المنافذ بالإضافة إلى عناوين IP.

بالنظر إلى المثال الموضح في الشكل 4-22، وبافتراض أن مستخدماً يعمل على الشبكة المنزلية من خلال المضيّف 10.0.0.1 يطلب صفحة ويب من خادم ويب (منفذ 80) بعنوان 128.119.40.186. يخصّص المضيّف 10.0.0.1 رقم منفذ (اعتباطي) للمصدر وليكن 3345، ويرسل وحدة البيانات إلى شبكة الاتصالات المحلية. يستلم موجّه NAT وحدة البيانات، ويولّد رقم منفذ جديد وليكن 5001 لمصدر وحدة البيانات تلك، ثم يستبدل عنوان IP للمصدر بعنوان IP لواجهته المتصلة بالشبكة واسعة النطاق أي 138.76.29.7، ويستبدل رقم منفذ المصدر الأصلي 3345 برقم منفذ المصدر الجديد 5001. عندما يولّد موجّه NAT رقماً جديداً لمنفذ المصدر يمكن أن يختار أي رقم غير موجود حالياً في جدول ترجمة NAT. (لاحظ أن حقل رقم المنفذ مكون من 16 بتاً ولذا يمكن أن يدعم بروتوكول NAT أكثر من 60 ألف توصيلة في نفس الوقت مع عنوان واحد لواجهة الموجّه المتصلة بالشبكة واسعة النطاق!). يضيف NAT الموجود في الموجّه أيضا مُدخلاً إلى جدول ترجمة NAT. لا يدرك خادم الويب أن رزمة البيانات الواصلة والتي تحتوي على طلب HTTP قد عولجت بموجّه NAT، ويردّ بإرسال رزمة بيانات تحتوي عنوان IP لموجّه NAT لعنوان الوجهة ورقم منفذ الوجهة 5001. عندما تصل وحدة البيانات هذه إلى موجّه NAT

يقوم الموجّه بالبحث في جدول ترجمة الـ NAT مستخدماً عنوان IP للوجهة ورقم منفذ الوجهة للحصول على عنوان IP المناسب (10.0.0.1) ورقم منفذ الوجهة (3345) للمتصفّح في الشبكة المنزلية. عندئذ يعيد الموجّه كتابة عنوان الوجهة لرزمة البيانات ورقم منفذ الوجهة، ويرسل رزمة البيانات إلى الشبكة المنزلية.

حظيت NAT بانتشار واسع في السنوات الأخيرة. لكننا يجب أن نذكر بأن العديد من المثاليين في محيط فريق عمل هندسة الإنترنت (IETF) يعترضون على NAT بصوت عالٍ. يعترضون أولاً لأنه من المفروض أن تُستعمل أرقام المنافذ لعنونة العمليات وليس لعنونة المضيفات (هذا الانتهاك يمكن أن يسبّب في الحقيقة مشاكل للخادمات التي تعمل على الشبكة المنزلية لأنه كما رأينا في الفصل الثاني تنتظر عمليات الخادم الطلبات القادمة لأرقام منافذ معروفة ومحددة). ويعترضون ثانياً لأنه من المفروض أن الموجّهات تعالج الرزم حتى الطبقة 3 فقط. والسبب الثالث لاعتراضهم أن بروتوكول NAT ينتهك ما يسمّى بقضية من طرف إلى طرف؛ أي أن المضيفات يجب أن تتكلم مباشرة مع بعضها البعض بدون تدخّل عقد لتعديل عناوين IP وأرقام المنافذ. ويعترضون رابعاً وأخيراً لأنه يجب استخدام IPv6 (راجع الجزء 4-4-4) للتغلب على مشكلة النقص في عناوين IP، بدلاً من تلك الحلول الترقيعية المؤقّتة للمشكلة كحلول NAT. لكن سواء شئنا أم أبينا أصبح الـ NAT مكوّناً مهماً للإنترنت.

من المشكلات الرئيسة الأخرى التي تواجه NAT تداخلها مع تطبيقات النظائر، كمشاركة النظائر للملفات وتطبيقات النظائر لنقل الصوت عبر الإنترنت. تذكّر من الفصل الثاني أنه في تطبيقات النظائر يستطيع أي نظير مشارك A أن يبدأ توصيلة TCP مع أي نظير مشارك آخر B. تكمن المشكلة في أنه لو كان النظير B وراء الـ NAT فإنه لا يستطيع العمل كخادم وبالتالي لا يستطيع أن يقبل توصيلات TCP. كما سنرى في مسائل الواجب المنزلي يمكن التخلص من مشكلة NAT هذه إذا لم يكن النظير A وراء الـ NAT. في هذه الحالة يمكن أن يتصل النظير A أولاً بالنظير B عن طريق نظير آخر C ليس وراء الـ NAT ويجري معه النظير B حالياً اتّصال TCP. يمكن أن يسأل النظير A النظير B عن طريق النظير C

أن يبدأ اتّصال TCP خلفي مباشرةً مع النظير A. بمجرد إنشاء الاتصال المباشر بين النظيرين A وB يمكن أن يتبادلا الرسائل أو الملفات. تسمّى هذه العملية "الاتّصال الخلفي" وتستخدم في الواقع من قبل الكثير من تطبيقات النظائر لتجاوز NAT. إذا كان كلٌّ من النظيرين A وB وراء الـ NAT الخاص به تكون هذه الحالة أصعب نوعاً ما لكن يمكن أن تعالج باستعمال تطبيقات الترحيل (relays) كما رأينا مع مرحّلات سكاي (Skye) في الفصل الثاني.

بروتوكول UPnP

يوفر بروتوكول UPnP (Universal Plug and Play) على نحو متزايد عبوراً للـ NAT وذلك بالسماح للمضيف باكتشاف وتهيئة الـ NAT القريب [UPnP Forum 2007]. يتطلّب UPnP أن يكون كلٌّ من المضيف والـ NAT متوافقين مع UPnP. وباستخدام UPnP يمكن أن يطلب تطبيقٌ ما يجري تشغيله على المضيف من الـ NAT الترجمة بين (عنوان IP ورقم المنفذ الخاصين به) و(عنوان IP العام ورقم المنفذ العام) عند توجيه طلب إلى رقم منفذٍ عامٍ ما. إذا قَبِل الـ NAT الطلب فيمكن للعُقد من الخارج أن تبدأ توصيلات TCP مع (عنوان IP العام ورقم المنفذ العام). وعلاوةً على ذلك يُمكّن UPnP التطبيق من معرفة قيمة (عنوان IP العام ورقم المنفذ العام) وبالتالي يمكن أن يعلنه التطبيق إلى العالم الخارجي.

كمثال افترض أن مضيفك وراء الـ NAT ويستعمل UPnP وله عنوان خاص 10.0.0.1 ويشغل تطبيق BitTorrent على منفذ 3345. وافترض أيضاً أن عنوان IP العام للـ NAT هو 138.76.29.7. من الطبيعي أن تطبيق BitTorrent لديك يحتاج أن يكون قادراً على قبول توصيلات من المضيفات الآخرى لكي يمكنه تبادل البيانات معهم. ولذلك يطلب تطبيق BitTorrent في مضيفك من الـ NAT تكوين "فتحة" لترجمة (10.0.0.1، 3345) إلى (138.76.29.7، 5001) (يختار التطبيق رقم المنفذ العام؛ في هذا المثال 5001). يمكن أن يعلن تطبيق BitTorrent في مضيفك أيضاً إلى مقتفيه بأنه موجود في العنوان (138.76.29.7، 5001). بهذا الأسلوب يمكن أن يتصل مضيف خارجي يشغل BitTorrent بالمقتفي ويعرف أن تطبيق BitTorrent

لديك موجود من خلال العنوان (138.76.29.7 ، 5001). وبالتالي يمكن أن يرسل المضيف الخارجي TCP SYN إلى العنوان (138.76.29.7 ، 5001). عندما يستلم NAT رزمة SYN سيغيّر عنوان IP ورقم منفذ الوجهة في الرزمة إلى (10.0.0.1 ، 3345) ثم يرسلها عبر الـ NAT.

الخلاصة هي أن UPnP يسمح للمضيفات الخارجية ببدء الاتصال مع مضيفات وراء الـ NAT باستعمال TCP أو UDP. لقد كان الـ NAT ولفترة طويلة عدواً لتطبيقات P2P؛ وقد وفر UPnP حلاً فعّالاً ومتيناً لاجتياز الـ NAT والذي ربما كان المنقذ لتلك التطبيقات. لقد كانت مناقشتنا هنا للـ NAT وUPnP مختصرة بالضرورة، وللمزيد من التفصيل راجع [Huston and UPnP 2004; Cisco NAT 2007].

4-4-3 بروتوكول رسائل التحكم في الإنترنت (ICMP)

تذكّر أن طبقة شبكة الإنترنت لها ثلاثة مكوّنات رئيسة: بروتوكول IP (نوقش في الجزء السابق)، وبروتوكولات التوجيه (وتضم RIP وOSPF وBGP) (وسوف نغطيها في الجزء 4-6)، وبروتوكول ICMP (وهو موضوع هذا الجزء).

تم وصف بروتوكول ICMP في [RFC 792]، وتستخدمه المضيفات والموجّهات لتبادل معلومات طبقة الشبكة فيما بينها. إن أكثر استعمالات بروتوكول ICMP هو للإبلاغ عن الخطأ. على سبيل المثال ربما صادفت رسالة خطأ مثل "لا يمكن الوصول لشبكة الوجهة" عند تشغيلك Telnet أو FTP أو HTTP. هذه الرسالة أصلها بروتوكول ICMP. في وقت ما قد لا يستطيع موجّه IP إيجاد مسار يصل إلى المضيف المحدّد عند تشغيلك تطبيق Telnet أو FTP أو HTTP. عندئذٍ ينشئ الموجّه رسالة ICMP من النوع 3 تتضمن وصفاً للخطأ الذي حدث ويرسلها إلى مضيفك.

غالباً ما يُعتبر بروتوكول ICMP جزءاً من بروتوكول IP، ولكنه من الناحية المعمارية يقع فوق IP مباشرة حيث إن رسائل ICMP يتم حملها داخل وحدات بيانات IP. بمعنى أن رسائل ICMP تمثل الحمل الآجر ضمن وحدات بيانات IP تماماً كما

تُحمل قطع بيانات TCP أو UDP في وحدات بيانات IP. وبنفس الطريقة عندما يستلم مضيف وحدة بيانات IP تحمل رسالة ICMP فإنه يقوم بانتزاع محتويات الرسالة تماماً كما يفعل مع قطع بيانات TCP أو UDP.

تحتوي رسالة ICMP على حقل يحدد النوع والكود لها، كما تتضمن ترويسة وحدة بيانات IP التي تسببت في توليد رسالة ICMP تلك في المقام الأول وأول ثماني بايتات منها (كي يتمكن المُرسِل من تحديد وحدة البيانات التي سببت الخطأ). يبين الشكل 4-23 بعض أنواع رسائل ICMP. لاحظ أن رسائل ICMP لا يقتصر استخدمها على الإبلاغ عن حالات الأخطاء.

الوصف	الكود	النوع
رد الصدى	0	0
لا يمكن الوصول لشبكة الوجهة	0	3
لا يمكن الوصول لمضيف الوجهة	1	3
لا يمكن الوصول لبروتوكول الوجهة	2	3
لا يمكن الوصول لمنفذ الوجهة	3	3
شبكة الوجهة غير معروفة	6	3
مضيف الوجهة غير معروف	7	3
خنق المصدر (التحكم في الازدحام)	0	4
طلب صدى	0	8
إعلان من موجّه	0	9
اكتشاف موجّه	0	10
انتهاء فترة TTL	0	11
ترويسة وحدة البيانات غير صحيحة	0	12

الشكل 4-23 أنواع رسائل بروتوكول ICMP.

يرسل برنامج البينج (ping) الشهير رسالة ICMP من النوع "8" بالكود "0" إلى المضيف المحدّد. يرد مضيف الوجهة الذي يرى رسالة "طلب الصدى" (echo request) برسالة ICMP "رد الصدى" (echo reply) من النوع "0" بكود "0". تدعم معظم نظم TCP/IP تحقيق خادم البينج كجزء مباشر من نظام التشغيل (أي أن الخادم ليس عملية (process)). يحتوي الفصل الأول من كتاب [Stevens 1990] على النص الأصلي لبرنامج زبون البينج. لاحظ أنه من الضروري أن يكون بوسع برنامج الزبون أن يطلب من نظام التشغيل توليد رسالة ICMP من النوع "8" بكود "0".

من رسائل ICMP الأخرى الشائقة رسالة خنق المصدر (source quench)، رغم أنها نادراً ما تستخدم حالياً. كان الغرض الأساسي من هذه الرسالة السماح لموجّه يعاني من الازدحام بإرسال تلك الرسالة إلى مضيف لإجباره على تخفيض معدل إرساله. لقد رأينا في الفصل الثالث أن بروتوكول TCP لديه آلية للتحكم في الازدحام تعمل في طبقة النقل بدون استعمال رسائل طبقة الشبكة لخنق المصدر.

قدّمنا في الفصل الأول برنامج متتبع المسار (Traceroute) والذي يسمح لنا بتتبّع المسار من مضيف معين إلى أي مضيف آخر في العالم. ومن الجدير بالذكر أن هذا البرنامج أيضاً يستخدم رسائل ICMP. لتقرير أسماء وعناوين الموجّهات بين المصدر والوجهة يقوم برنامج تتبع المسار بإرسال سلسلة من وحدات بيانات IP العادية إلى الوجهة. تحمل كلٌّ من هذه الوحدات قطعة UDP برقم منفذ UDP غير محتمل الوجود. ويكون زمن TTL في أول هذه الوحدات له القيمة 1، وفي الثانية له القيمة 2، وفي الثالثة له القيمة 3، وهكذا. يبدأ المصدر أيضاً موقّتات لكل وحدة من وحدات البيانات. عندما تصل وحدة البيانات n إلى الموجّه n يلاحظ الموجّه n أن مدة TTL لوحدة البيانات قد انتهت. وفقاً لقواعد بروتوكول IP يتخلص الموجّه من وحدة البيانات ويرسل رسالة ICMP تحذيرية إلى المصدر (من النوع 11 بكود 0) تتضمّن اسم الموجّه وعنوان IP له. عندما تصل تلك الرسالة إلى المصدر يحصل على زمن رحلة الذهاب والإياب من الموقّت واسم وعنوان IP للموجّه n من رسالة ICMP.

نبذة عن الأمن (Focus on Security)

تفتيش وحدات البيانات: برامج الحماية وأنظمة اكتشاف الاختراق

لنفترض أنك تضطلع بمهمّة إدارة شبكة في البيت أو القسم أو الجامعة أو الشركة. من السهل على المهاجمين الذين يعرفون حيز العناوين لتلك الشبكة إرسال وحدات بيانات IP إلى أيٍّ من تلك العناوين. يمكن أن تقوم وحدات البيانات تلك بأي نوع من الأشياء المخادعة مثل رسم مخطط لشبكتك عن طريق ما يسمى بمسح البينج (ping sweeps) ومسح المنافذ (port scans)، وتخريب المضيفات الضعيفة برزم مشوّهة، وإغراق الخادمات بفيضان من رسائل ICMP، وإصابة المضيفات بتضمين برمجيات خبيثة (malware) في الرزم.

بصفتك المشرف على الشبكة ما الذي يمكنك فعله إزاء كل أولئك الأشرار القادرين على إرسال رزم خبيثة إلى شبكتك؟ هناك آليتان شائعتان لصد هجمات الرزم الخبيثة: برامج الحماية (firewalls) وأنظمة اكتشاف الاختراق (IDSs) (Intrusion Detection Systems).

قد تحاول أولاً تركيب برامج الحماية بين شبكتك والإنترنت (معظم موجّهات الوصول (access routers) اليوم مزودة ببرامج الحماية). تقوم برامج الحماية بتفتيش ترويسات وحدات وقطع البيانات، وتمنع وحدات البيانات المريبة من الدخول إلى الشبكة الداخلية. على سبيل المثال قد تُعدّ برامج الحماية لمنع كل رزم رسائل ICMP لطلبات الصدى، وبذلك تمنع المهاجمين من عمل مسح بينج لحيز العناوين لشبكتك. يُمكن أيضاً أن تمنع برامج الحماية الرزم بناءً على عناوين بروتوكول الإنترنت للمصدر والوجهة وأرقام المنافذ. كما يُمكن أن تُعدّ برامج الحماية لتعقب توصيلات TCP والسماح فقط بدخول وحدات البيانات التي تنتمي إلى التوصيلات المسموح لها.

مكن توفير حماية إضافية باستخدام نظام IDS، والذي عادةً ما يركّب على تخوم (حدود) الشبكة ليقوم بتفتيش "أعمق" للرزم وذلك بفحص ليس فقط حقول الترويسة ولكن أيضاً بيانات الحمل الآجر في وحدة البيانات (بما في ذلك بيانات طبقة البرامج). يحتوي نظام IDS على قاعدة بيانات لتوقيعات الرزم (packet signature) المعروفة بأنها تشكل جزءاً من هجمات. يتم تحديث قاعدة البيانات تلك آلياً كلما اكتشفت هجمات جديدة. بينما تعبر الرزم نظام IDS يحاول النظام مطابقة حقول الترويسة وحمولات تلك الرزم بالتوقيعات الموجودة في قاعدة البيانات. إذا عثر على تطابق يقوم IDS بإصدار إنذار. وهناك أيضاً أنظمة لمنع الاختراق (IPS) وهي تشبه أنظمة IDS إلا أنها تمنع تلك الرزم من دخول الشبكة بالإضافة إلى إصدار الإنذارات. سنستعرض في الفصل الثامن تفاصيل أكثر حول برامج الحماية ونظم IDS.

هل بإمكان برامج الحماية ونظم IDS حماية شبكتك حمايةً كاملةً من كل الهجمات؟ من الواضح أن الجواب لا، لأن المهاجمين يقومون بشكلٍ مستمر بهجمات جديدة ليست لها أية توقيعات متوفرة في قاعدة البيانات. لكن برامج الحماية ونظم IDS التقليدية المعتمدة على التوقيعات تفيد بلا شك في حماية شبكتك من الهجمات المعروفة.

لكن كيف يعرف برنامج متتبع المسار متى يجب التوقّف عن إرسال قطع UDP؟ تذكّر أن البرنامج يقوم بزيادة مدة TTL لكل وحدة بيانات يرسلها، وبالتالي ستقطع في النهاية إحدى وحدات البيانات الطريق بطوله إلى مضيف الوجهة. ولأن وحدة البيانات تلك تحتوي على قطعة UDP برقم منفذ غير محتمل الوجود فإن مضيف الوجهة سيرسل رسالة ICMP (من النوع 3 بكود 3) إلى المصدر للإبلاغ عن أن منفذ UDP في وحدة البيانات بعيد المنال. عندما يستلم مضيف المصدر رسالة ICMP تلك، يعرف بأنه ليس بحاجة إلى أن يرسل وحدات بيانات اختبار إضافية (في الحقيقة يرسل برنامج متتبع المسار القياسي مجموعات من ثلاث وحدات بيانات لها نفس مدة TTL، وهكذا يتكون مخرج متتبع المسار من ثلاث قيم لكل TTL).

بهذه الطريقة يُحدِّد مضيف المصدر عدد وهويّات الموجّهات التي تقع بينه وبين مضيف الوجهة وكذلك مدة رحلة الذهاب والإياب بينهما. لاحظ أن برنامج زبون متتبع المسارات يجب أن يكون قادراً على الإيعاز إلى نظام التشغيل بتوليد وحدات بيانات UDP بقيم TTL معينة، ويجب أيضاً أن يكون قادراً على تلقي إخطارات من نظام التشغيل عندما تصل رسائل ICMP. الآن وبعد أن عرفت كيف يعمل برنامج متتبع المسارات، قد تريد معاودة تشغيله والتمرن عليه أكثر.

4-4-4 بروتوكول IPv6

في أوائل التسعينيات بدأ فريق عمل هندسة الإنترنت (IETF) محاولة لتطوير بروتوكول يخلف بروتوكول IPv4. كان أحد الحوافز الأساسية لهذا الجهد هو إدراك أن فضاء العناوين المكونة من 32 بتاً يشرف على النفاذ بمعدل سريع، مع توصيل شبكات فرعية وعُقد بروتوكول IP جديدة بالإنترنت (وتخصيص عناوين IP فريدة) بمعدّلات عالية للغاية. لتلبية هذه الحاجة لفضاء كبير لعناوين IP تم تطوير بروتوكول جديد للإنترنت IPv6. وانتهز مصممو IPv6 الفرصة أيضاً لتحسين وتوسيع الإمكانيات الأخرى لبروتوكول IPv4 استناداً على الخبرة التشغيلية المتراكمة مع ذلك البروتوكول.

كان الموضوع الذي دار عليه نقاش كبير هو متى سيتم نفاذ عناوين IPv4 بالكامل (وبالتالي لا يُمكن توصيل شبكات فرعية جديدة بالإنترنت)؟ كانت تخمينات زعيمي مجموعة توقعات عمر العناوين في فريق عمل هندسة الإنترنت هو أنها ستنفذ في عامي 2008 و2018، على التوالي [Solensky 1996]. في عام 1996 أعلن مكتب التسجيل الأمريكي لأعداد الإنترنت (ARIN) أن كل عناوين الفئة A لبروتوكول IPv4 قد خصّصت، وأن 62 بالمائة من عناوين الفئة B قد خصّصت، وأن 37 بالمائة من عناوين الفئة C قد خصّصت [ARIN 1996]. ويمكنك الاطلاع على تقرير حديث عن تخصيص عناوين IPv4 في [Hain 2005]. رغم أن هذه التقديرات والإحصاءات تشير إلى أنه لا يزال هناك متسع من الوقت حتى تُستنزف عناوين IPv4، إلا أنه تم أيضاً إدراك أن وقتاً طويلاً سيكون مطلوباً لانتشار تقنية جديدة على مثل هذا النطاق الواسع؛ ولذا بدأ العمل في بروتوكول الإنترنت القادمة IPng [Bradner 1996; RFC 1752]. كانت نتيجة هذا الجهد وضع مواصفات نسخة بروتوكول الإنترنت 6 (IPv6) [RFC 2460]. غالباً ما يطرح هنا السؤال التالي: "وماذا حدث لبروتوكول IPv5"؟ كان التصوّر الأولي بأن بروتوكول ST-2 سيصبح IPv5، لكن ST-2 أسقط لاحقاً لصالح بروتوكول RSVP والذي سنناقشه في الفصل السابع.

من مصادر المعلومات الممتازة حول IPv6 صفحة الويب الخاصة بالجيل القادم من بروتوكول الإنترنت IPng [Hinden 2007] وكتاب هويتيما حول هذا الموضوع [Huitema 1998].

صيغة وحدة بيانات بروتوكول IPv6

يبين الشكل 4-24 صيغة وحدة بيانات IPv6. ومن أهم التغييرات في بروتوكول IPv6 والتي تتضح من صيغة وحدة البيانات:

- التوسع في العناوين: زاد بروتوكول IPv6 حجم عنوان بروتوكول الإنترنت من 32 بتاً إلى 128 بتاً. وهذا يضمن بأن العالم لن يستنفذ عناوين بروتوكول الإنترنت (فيمكن الآن لكل حبة رمل على كوكب الأرض أن يكون لها

عنوان IP!). بالإضافة إلى عناوين الإرسال الفردي (unicast) وعناوين الإرسال الجماعي (multicast)، يوفر IPv6 نوعاً جديداً من العناوين يسمّى عنوان (anycast) والذي يسمح بتوصيل وحدة البيانات إلى أي مضيف ضمن مجموعة من المضيفات. (يمكن استخدام هذه الميزة على سبيل المثال لإرسال تعليمة "GET" في بروتوكول HTTP إلى الخادم الأقرب في عدد من مواقع الويب البديلة التي تحتوي على وثيقةٍ ما).

32 بتاً

رقم الإصدار | نوع حركة المرور | وسمة التدفق
طول الحمل الآجر | الترويسة التالية | حد القفزات
عنوان المصدر (128 بتاً)
عنوان الوجهة (128 بتاً)
البيانات (الحمل الآجر)

الشكل 4-24 صيغة وحدة بيانات IPv6.

- انسيابية الترويسة المكونة من 40 بايتاً: كما سنتناول لاحقاً، تم إسقاط عدد من حقول IPv4 أو جعلها اختيارية. من شأن استخدام الترويسة الجديدة بطول ثابت قدره 40 بايتاً تسريع معالجتها. كما يسمح استخدام نظام تكويد جديد للخيارات بمرونة أكثر في معالجة تلك الخيارات.
- وسم التدفق والأولوية: يستخدم IPv6 تعريفاً مبهماً بعض الشيء للتدفق، حيث ذكر في RFC 1752 وRFC 2460 أن هذا يسمح بـ "وسم الرزم التي تنتمي إلى تدفقات معينة والتي يطلب المُرسِل معالجة خاصة لها كنوعية غير معتادة من الخدمة أو خدمة فورية". على سبيل المثال قد يُعامَل إرسال الفيديو والتسجيل الصوتي كتدفق، بينما لا تُعد التطبيقات الأكثر تقليدية – كإرسال الملفات والبريد الإلكتروني – تدفقاً. من الممكن معاملة حركة

مرور البيانات من قِبَل مستخدم له أولوية عالية كتدفق (كما في حالة شخص ما يدفع ثمناً أعلى لخدمة أفضل لحركة مرور بياناته). ومع ذلك فمن الواضح أن مصممي IPv6 يتوقعون ضرورة توفير القدرة على التمييز بين أنواع التدفق المختلفة، حتى وإن كان المعنى الدقيق لـ "تدفق" لم يحدد بعد. يوجد أيضاً في ترويسة IPv6 حقلٌ مكون من 8 بتات لتحديد نوع حركة مرور البيانات. يمكن استخدام هذا الحقل – مثل حقل TOS في IPv4 – لإعطاء أولوية لبعض وحدات البيانات ضمن تدفق معين، أو لإعطاء أولوية لوحدات البيانات من بعض التطبيقات (مثل ICMP) على وحدات البيانات من التطبيقات الأخرى (مثل شبكات الأخبار).

كما لاحظنا سابقاً تكشف المقارنة بين الشكل 4-24 والشكل 4-13 بساطة وانسيابية أكثر لتركيب وحدة بيانات IPv6. تعرّف الحقول التالية في بروتوكول IPv6:

- رقم النسخة (الإصدار): يُميّز هذا الحقل المؤلف من 4 بتات رقم نسخة بروتوكول الإنترنت. وليس مستغرباً أن يحتوي هذا الحقل على القيمة 6 لبروتوكول IPv6. لاحظ أن وضع القيمة 4 في هذا الحقل لا يُكوِّن وحدة بيانات IPv4 صحيحة. (لو حدث ذلك لكانت الحياة أسهل بكثير!، راجع المناقشة الخاصة بالانتقال من IPv4 إلى IPv6 فيما بعد).
- نوع حركة مرور البيانات: يماثل هذا الحقل المكون من 8 بتات – من حيث المبدأ – حقل TOS الموجود في IPv4.
- وسمة التدفق (flow label): كما ناقشنا سابقاً يُستعمَل هذا الحقل المكون من 20 بتاً لتمييز تدفق من وحدات البيانات.
- طول الحمل الآجر: تعامل هذه القيمة المكونة من 16 بتاً كعدد صحيح بدون إشارة، وذلك لتحديد عدد البايتات في وحدة بيانات IPv6 التي تلي الترويسة ثابتة الطول والمكونة من 40 بايتاً.

- الترويسة التالية (next header): يميّز هذا الحقل البروتوكول الذي ستسلم إليه محتويات وحدة البيانات تلك (مثلاً TCP أو UDP). يستخدم هذا الحقل قيماً مماثلة لتلك المستخدمة في IPv4.
- الحد الأعلى لعدد القفزات (hop limit): تخفض محتويات هذا الحقل بمقدار واحد عند كل موجّه يقوم بإرسال وحدة البيانات تلك. فإذا أصبحت قيمته صفراً، سيهمل الموجّه وحدة البيانات ولا يرسلها.
- عناوين المصدر والوجهة: يوجد وصف للصيغ المختلفة لعناوين IPv6 والمكونة من 128 بتاً في RFC 4291.
- البيانات: يمثل هذا الجزء "الحمل الآجر" لوحدة بيانات IPv6. عندما تصل وحدة بيانات إلى وجهتها يتم استخلاص هذا الجزء من وحدة البيانات ونقله إلى البروتوكول المحدد في حقل "الترويسة التالية".

حددت المناقشة السابقة الغرض من الحقول المتضمنة في وحدة بيانات IPv6. بمقارنة صيغة وحدة بيانات IPv6 في الشكل 4-24 مع صيغة وحدة بيانات IPv4 التي رأيناها في الشكل 4-13 سنلاحظ أن عدّة حقول في وحدة بيانات IPv4 لم تعد موجودة في وحدة بيانات IPv6:

- التجزئة وإعادة التجميع لوحدات البيانات: لا يسمح IPv6 بتجزئة وإعادة تجميع وحدات البيانات في الموجّهات المتوسّطة، وإنما يمكن أن تؤدَّى هذه العمليات فقط بواسطة المصدر والوجهة. إذا استلم موجّه وحدة بيانات IPv6 كبيرة جداً لكي ترسل على الوصلة الخارجة، فإن الموجّه ببساطة يسقط وحدة البيانات ويرسل رسالة خطأ ICMP "رزمة كبيرة جداً" إلى المُرسِل. يمكن حينئذ أن يعيد المُرسِل إرسال البيانات مستخدماً حجماً أصغر لوحدة بيانات IP. إن التجزئة وإعادة التجميع عملية مضيِّعة للوقت، لذا ستؤدي إزالة هذه الوظيفة من الموجّهات ووضعها مباشرة في الأنظمة الطرفية إلى تسريع بروتوكول الإنترنت إلى حدٍّ كبير.
- المجموع التدقيقي للترويسة: نظراً لأن بروتوكولات طبقة النقل (مثل TCP وUDP) وطبقة ربط البيانات (مثل الإيثرنت) في رصّة بروتوكولات الإنترنت

تقوم بتدقيق البيانات، شعر مصممو بروتوكول الإنترنت الجديد بأن هذه الوظيفة في طبقة الشبكة زائدة ويمكن إزالتها. مرة أخرى كانت المعالجة السريعة لرزم بروتوكول الإنترنت محط اهتمام المصممين. تذكّر من مناقشتنا لـ IPv4 في الجزء 4-4-1 أنه يلزم حساب المجموع التدقيقي للترويسة بعد كل قفزة نظراً لوجود حقل TTL والذي تتغيّر قيمته مع كل قفزة. كما هو الحال مع التجزئة وإعادة التجميع كانت هذه العملية مكلّفة في بروتوكول IPv4.

- الخيارات: لم يعد حقل الخيارات جزءاً من ترويسة IP المعيارية. ومع ذلك فإنه لم يُلغ تماماً وإنما أصبح حقل الخيارات أحد الترويسات التالية المحتملة والمشار إليها من ترويسة IPv6. أي مثلما يمكن أن تكون ترويسة TCP أو UDP الترويسة التالية ضمن حزمة بروتوكول الإنترنت كذلك يمكن أن يكون حقل الخيارات. يؤدّي إزالة حقل الخيارات إلى جعل ترويسة بروتوكول الإنترنت ثابتة الطول ومؤلفة من 40 بايتاً.

تذكّر من مناقشتنا في الجزء 4-4-3 أن بروتوكول ICMP يُستخدم من قِبَل عُقد بروتوكول الإنترنت للإبلاغ عن حالات الخطأ وتزويد النظام الطرفي بمعلومات محدودة (على سبيل المثال رسالة رد الصدى لرسالة البينج). لقد تم تعريف نسخة جديدة من بروتوكول ICMP لبروتوكول IPv6 في RFC 4443. بالإضافة إلى إعادة تنظيم تعريفات الأنواع والأكواد الموجودة، أضاف ICMPv6 أيضاً أنواعاً وأكواداً جديدة تطلّبتها وظائف IPv6 الجديدة. يشمل ذلك "رزمة كبيرة جداً"، و"خيارات IPv6 غير معروفة". كما يتضمّن ICMPv6 وظائف بروتوكول IGMP والتي سندرسها في الجزء 4-7. في السابق كان IGMP – والذي يستعمل لإدارة انضمام مضيف ومغادرته لمجموعات الإرسال الجماعي – بروتوكولاً منفصلاً عن ICMP في IPv4.

الانتقال من IPv4 إلى IPv6

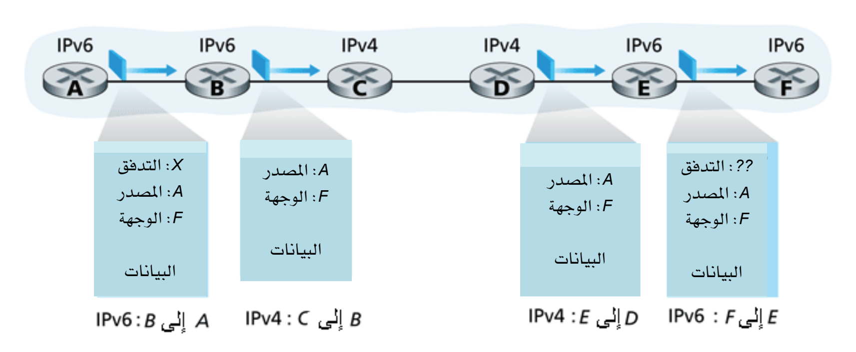
دعنا الآن بعد أن رأينا تفاصيل تقنية IPv6 النظر في مسألة عملية جداً: كيف ستتحول الإنترنت العامّة والتي تعمل طبقاً لبروتوكول IPv4 إلى بروتوكول IPv6؟ تكمن المشكلة في أنه بالرغم من توافق أنظمة IP الجديدة للعمل مع بروتوكول IPv4 (أي يمكنها إرسال وتوجيه واستقبال وحدات بيانات IPv4) إلا أن أنظمة IPv4 المستخدمة حالياً لا يمكنها معالجة وحدات بيانات IPv6. هناك عدّة خيارات ممكنة.

أحد تلك الخيارات هو الإعلان عن موعد محدد يتم فيه توقف تام للإنترنت وترقية كل أجهزتها من IPv4 إلى IPv6. ولقد كان آخر تحول هام في تقنية الإنترنت ما حدث منذ ما يقرب من 25 سنة تقريباً للإنتقال من بروتوكول NCP إلى بروتوكول TCP لخدمة النقل الموثوق فيه للبيانات. وحتى في ذلك الوقت [RFC 801] عندما كانت الإنترنت صغيرة جداً وتدار بواسطة عدد صغير من البرامج المساعدة (wizards) كان تحديد يومٍ بعينه لحدوث تحول في التقنية غير ممكن. هذا الأمر مستحيل بدرجة أكبر اليوم لأنه يتضمّن مئات الملايين من الأجهزة والملايين من مشرفي الشبكات ومستخدميها. يقدم RFC 4213 وصفاً لطريقتين (يمكن اتباعهما منفردتين أو معاً) للإحلال التدريجي لمضيفات وموجّهات IPv6 ضمن عالم IPv4 (بالطبع مع الهدف بعيد المدى لتحويل كل عقد IPv4 في النهاية إلى IPv6).

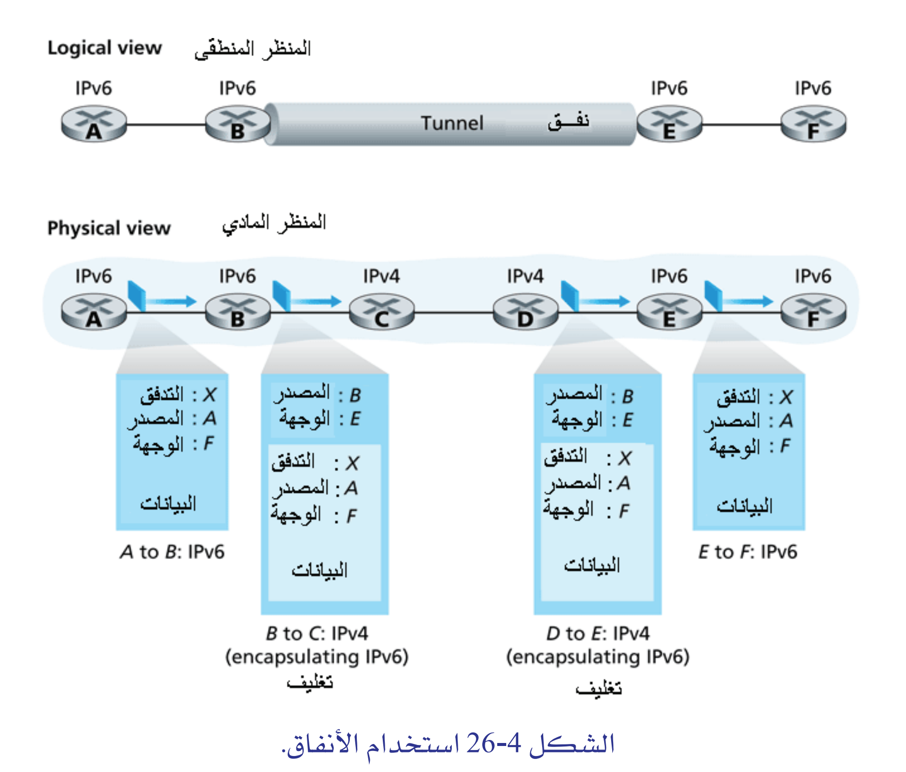
ربما تكون الطريقة الأبسط هي تقديم عقد IP برصّة بروتوكولات مزدوجة، حيث تتضمن عقد IPv6 تحقيقاً لبروتوكول IPv4 أيضاً. تدعى مثل هذه العقدة عقدة IPv6/IPv4 في RFC 4213، ولها القدرة على إرسال واستلام وحدات بيانات لكلٍّ من IPv4 وIPv6. عند توصيل عقد IPv6/IPv4 مع عقد IPv4 تستخدم وحدات بيانات IPv4، وعند توصيلها مع عقد IPv6 تستخدم وحدات بيانات IPv6. يجب أن يكون لعقد IPv6/IPv4 عناوين IPv6 وIPv4. وعلاوةً على ذلك يجب أن تكون قادرة على تحديد ما إذا كانت عقدة أخرى قادرة على التعامل بكلٍّ من IPv6 وIPv4 أو بـ IPv4 فقط. هذه المشكلة يمكن أن تحل عن طريق DNS (راجع الفصل الثاني)،

والذي يمكن أن يُرجِع عنوان IPv6 إذا كانت العقدة المعطى اسمها قادرة على التعامل ببروتوكول IPv6 أو يُرجِع عنوان IPv4 فيما عدا ذلك. بالطبع يُرجِع DNS عنوان IPv4 فقط إذا كانت العقدة التي تصدر طلب DNS قادرة فقط على استعمال IPv4.

في طريقة رصّة البروتوكولات المزدوجة، إذا كان المُرسِل أو المُستقبِل قادراً على التعامل ببروتوكول IPv4 فقط فإنه يجب استخدام وحدات بيانات IPv4. ونتيجة لذلك قد يصل الأمر في النهاية إلى أن عقد IPv6 ترسل وحدات بيانات IPv4 إلى بعضها البعض. هذه الحالة موضحة في الشكل 4-25. افترض أن العقدة A قادرة على التعامل ببروتوكول IPv6 وتريد إرسال وحدة بيانات إلى العقدة F والتي تعمل أيضاً ببروتوكول IPv6. يمكن أن تتبادل العقد A وB وحدات بيانات IPv6. ومع ذلك يجب أن تكوّن العقدة B وحدة بيانات IPv4 للإرسال إلى C. بالتأكيد يمكن أن ينسخ حقل البيانات من وحدة بيانات IPv6 إلى حقل بيانات وحدة بيانات IPv4 مع عمل تحويل للعناوين. ومع ذلك فعند التحويل من IPv6 إلى IPv4 سيكون هناك حقول معينة في وحدة بيانات IPv6 (مثل حقل معرّف التدفق) ليس لها نظير في IPv4 وبالتالي ستفقد المعلومات الموجودة في تلك الحقول. وهكذا بالرغم من أن E وF يمكن أن يتبادلا وحدات بيانات IPv6 إلا أن وحدات بيانات IPv4 الواصلة لـ E من D لا تحتوي على كل الحقول التي كانت في وحدة بيانات IPv6 الأصلية التي أرسلت من A.



الشكل 4-25 أسلوب رصة البروتوكولات المزدوجة.



الشكل 4-26 استخدام الأنفاق.

هناك طريقة بديلة لطريقة رصّة البروتوكولات المزدوجة تم مناقشتها أيضاً في RFC 4213، وتعرف باستخدام الأنفاق (tunnels). يمكن أن يحل استخدام الأنفاق المشكلة المذكورة سابقاً وذلك بالسماح لـ E على سبيل المثال باستلام وحدة بيانات IPv6 مرسلة من قِبَل A. الفكرة الأساسية وراء استخدام الأنفاق هي كالتالي: افترض أن عُقدتي IPv6 (مثلاً B وE في الشكل 4-25) تريدان استخدام وحدات بيانات IPv6 لكن الموجهات الموصلة بينهما تستخدم IPv4. سنشير لمجموعة موجّهات IPv4 الموصلة بين اثنين من موجّهات IPv6 كـ "نفق" كما هو موضح في الشكل 4-26. باستخدام الأنفاق تأخذ عقدة IPv6 على جانب الإرسال للنفق (على سبيل المثال B) وحدة بيانات IPv6 كاملة وتضعها في حقل البيانات (الحمل الآجر) لوحدة بيانات IPv4، ثم تعنون وحدة بيانات IPv4 هذه إلى عقدة IPv6 على جانب الاستلام للنفق (على سبيل المثال E)، وترسل إلى العقدة الأولى في النفق (على سبيل المثال C). تقوم موجّهات IPv4 الفاصلة على طول النفق بتوجيه وحدة بيانات IPv4 هذه بينها

تماماً مثلما تفعل مع أي وحدة بيانات أخرى دون أن تدرك أن وحدة بيانات IPv4 تلك تحتوي على وحدة بيانات IPv6. في النهاية تستلم عقدة IPv6 على جانب الاستقبال للنفق وحدة بيانات IPv4 (حيث إنها تمثل وجهة وحدة بيانات IPv4)، وتدرك أن وحدة بيانات IPv4 تحتوي على وحدة بيانات IPv6، فتنتزع وحدة بيانات IPv6 وتقوم بتوجيه وحدة بيانات IPv6 تماماً كما تفعل عندما تستلم وحدة بيانات IPv6 من أحد جيرانها الذين يتعاملون ببروتوكول IPv6 مباشرة.

ننهي هذا الجزء بملاحظة أنه بينما كان الشروع في تبني IPv6 بطيئاً في بداية الأمر [Lawton 2001] إلا أنه قد أخذ في التسارع مؤخراً. وقد طلب المكتب الأمريكي للإدارة والميزانية (OMB) التحول إلى IPv6 بحلول شهر يونيو/حزيران 2008. يعطي انتشار الأجهزة كهواتف الإنترنت والأجهزة النقّالة الأخرى دفعاً إضافياً لانتشار أوسع لـ IPv6. ولقد حدد برنامج شراكة جيل أوروبا الثالث IPv6 كأسلوب معياري للعنونة للوسائط المتعددة النقالة [3GPP 2007]. حتى مع عدم انتشار IPv6 على نحو واسع في السنوات العشر الأولى من حياته القصيرة إلا أن الدعوة قوية الآن لاعتماده على المدى البعيد. إن نظام أرقام الهواتف المستعمل اليوم قد أخذ عدّة عقود للتطبيق، ولكنه مطبّق الآن لما يقرب من نصف قرن وبدون ما ينم عن الحاجة لتغييره. بنفس الطريقة قد يستغرق IPv6 بعض الوقت للسيطرة، ولكنه أيضاً قد يبقى لمدة طويلة فيما بعد. يقول براين كاربينتر ـ الرئيس السابق لمجلس بنية الإنترنت [IAB 2007] ومؤلف عدّة RFCs متعلّقة بـ IPv6 – "نظرت دائماً إلى هذا الأمر على أنه عملية مدتها 15 سنة تبدأ من عام 1995" [Lawton 2001]. وعلى حسب كلام كاربينتر نكون قد اقترينا من ثلاثة أرباع المدة!

من الدروس الهامة التي يمكن أن نتعلّمها من تجربة IPv6 أنه من الصّعب جداً تغيير بروتوكولات طبقة الشبكة. منذ أوائل التسعينيات أُعلِنت العديد من البروتوكولات الجديدة لطبقة الشبكة على أنها ستحدث انقلاباً كبيراً في الإنترنت لكن أغلب هذه البروتوكولات لاقت قبولاً محدوداً حتى الآن. من بين هذه البروتوكولات IPv6، وبروتوكولات الإرسال المتعدد (الجزء 4-7)، وبروتوكولات حجز الموارد (الفصل السابع). في الحقيقة يشبه تغيير بروتوكولات طبقة الشبكة

استبدال أساسات البيت (فمن الصعب إجراء ذلك بدون هدم للبيت بالكامل أو على الأقل نقل سكّان البيت بشكلٍ مؤقت). على الجانب الآخر شهدت الإنترنت استخدامات سريعة لبروتوكولات جديدة في طبقة التطبيقات. من الأمثلة التقليدية لذلك بالطبع الويب والرسائل الفورية ومشاركة النظائر للملفات. تتضمّن الأمثلة الأخرى التسجيل الصوتي وعرض الفيديو والألعاب الموزعة. يشبه تقديم بروتوكولات جديدة في طبقة التطبيقات إضافة طبقة جديدة من الطلاء إلى البيت والتي من السّهل نسبياً عملها (كما أنك لو اخترت لوناً جذاباً قد يقلدك آخرون في الحي). الخلاصة أنه يمكن أن نرى في المستقبل تغييرات في طبقة شبكة الإنترنت؛ لكن من المحتمل أن هذه التغييرات ستحدث على فترة زمنية أبطأ بكثير من التغييرات التي تحدث في طبقة التطبيقات.

4-4-5 رحلة قصيرة مع أمن بروتوكول IP

غطى الجزء 4-4-3 بعض تفاصيل بروتوكول IPv4 بما في ذلك الخدمات التي يوفرها وكيفية تحقيقها. وخلال قراءتك لذلك الجزء ربما لاحظت أنه لم يرد ذِكر أي خدمات للأمن (security). في الحقيقة صُمّم IPv4 في عصر (السبعينيات) عندما كان استعمال الإنترنت محصوراً بين باحثي الشبكات الموثوق فيهم والمؤتمنين فيما بينهم. كانت عملية بناء شبكة حاسب تتضمن العديد من تقنيات طبقة ربط البيانات تمثل تحدياً كافياً دون الحاجة للقلق حول هواجس الأمن.

لكن اليوم أصبح الأمن من القضايا الرئيسة، وانتقل باحثو الإنترنت لتصميم بروتوكولات جديدة لطبقة الشبكة توفر تشكيلة من خدمات الأمن. أحد هذه البروتوكولات الشائعة هو IPSec، والذي انتشر أيضاً على نحو واسع في الشبكات الخاصة الافتراضية ((Virtual Private Networks (VPNs)). وبالرغم من أن تغطية تفاصيل IPSec وعمليات التشفير التي يعتمد عليها ستأتي في الفصل الثامن إلا أننا سنعطي هنا مقدمة مختصرة عن خدمات IPSec.

لقد تم تصميم IPSec ليكون متوافقاً مع IPv4 وIPv6. وبالتحديد لكي نجني فوائد IPSec لسنا بحاجة إلى أن نستبدل رصّة البروتوكولات في كل الموجّهات

والمضيفات في الإنترنت. على سبيل المثال عند استعمال نمط النقل (transport mode) (أحد نمطين يوفرهما IPSec)، إذا أراد مضيفان الاتصال بشكلٍ آمن فمن الضروري أن يكون IPSec متوفراً فقط في هذين المضيفين. أما كل الموجّهات والمضيفات الأخرى فيمكن أن تستمر في استعمال IPv4 العادي.

وللدقة سنركّز هنا على نمط النقل لبروتوكول IPSec. في هذا النمط يؤسّس مضيفان جلسة IPSec أولاً فيما بينهما (ومن ثم فإن IPSec بروتوكول توصيلي). وأثناء تلك الجلسة تتمتع كل قطع بيانات TCP وUDP المرسلة بين المضيفين بخدمات الأمن المتوفرة من قِبَل IPSec. على جانب الإرسال تمرّر طبقة النقل القطعة إلى IPSec. يقوم IPSec بتشفير القطعة وإلحاق حقول الأمن الإضافية بها، ثم تغلّيف الحمل الآجر الناتج في وحدة بيانات IP عادية. (وفي الحقيقة فإن الأمر أكثر تعقيداً بعض الشيء من هذا كما سنرى في الفصل الثامن). بعد ذلك يرسل مضيف المصدر وحدة البيانات إلى الإنترنت لنقلها إلى مضيف الوجهة. وهناك يقوم IPSec بفك التشفير واسترجاع القطعة الأصلية وتمريرها إلى طبقة النقل.

تشمل الخدمات التي يوفرها IPSec ما يلي:

- الاتفاق على آليات التشفير: تسمح للمضيفين المتصلين بالاتفاق على خوارزميات ومفاتيح التشفير.
- تشفير الحمل الآجر لوحدة بيانات IP: عندما يستلم مضيف الإرسال قطعة من طبقة النقل يقوم IPSec بتشفير الحمل الآجر، ولا يمكن أن يُفك التشفير إلا من قِبَل IPSec في مضيف الاستقبال.
- سلامة البيانات (data integrity): يسمح بروتوكول IPSec لمضيف الاستقبال بالتحقق من عدم تعديل حقول الترويسة والحمل الآجر المشفر في وحدة البيانات أثناء رحلتها من المصدر إلى الوجهة.
- توثيق المصدر (origin authentication): عندما يستلم مضيف وحدة بيانات IPSec من مصدر موثوق به (لديه مفتاح تشفير موثوق فيه كما سنرى في الفصل الثامن) يطمئن المضيف بأن عنوان بروتوكول الإنترنت للمصدر في وحدة البيانات هو المصدر الفعلي لوحدة البيانات.

عندما يؤسّس مضيفان جلسة IPSec بينهما يتم تشفير وتوثيق جميع قطع بيانات TCP وUDP المرسلة بينهما. يزوّد IPSec تغطية عامّة لتأمين جميع الاتصالات بين المضيفين لكل تطبيقات الشبكة.

يمكن أن تستخدم شركةٌ ما بروتوكول IPSec للاتصال بشكلٍ آمن بالإنترنت العامّة غير الآمنة. ولتوضيح ذلك دعنا ننظر إلى مثال بسيط هنا. تصوّر شركة لديها عدد كبير من مندوبي المبيعات الجوّالين ومع كل واحد منهم حاسب نقال من الشركة. افترض أن مندوبي المبيعات يحتاجون للرجوع إلى معلومات حسّاسة جداً عن الشركة (كمعلومات عن المنتجات والأسعار) والمخزنة على خادم في مقر الشركة. افترض كذلك أن مندوبي المبيعات يحتاجون أيضاً لإرسال وثائق حسّاسة إلى بعضهم البعض. كيف يتم ذلك من خلال IPSec؟ والجواب أننا نركّب IPSec في الخادم وفي جميع حاسبات مندوبي المبيعات النقالة. وبذلك حينما يحتاج مندوب مبيعات للاتصال بالخادم أو مع مندوب آخر يتم إنشاء جلسة اتصال آمنة بينهما.

5-4 خوارزميات التوجيه (Routing Algorithms)

تعرضنا حتى الآن في هذا الفصل في الغالب لوظيفة التمرير في طبقة الشبكة. وعرفنا أنه عندما تصل رزمة إلى موجّه فإنه يبحث في جدول التمرير لديه ليحدد الوصلة التي ستوجّه إليها الرزمة. وعرفنا أيضاً أن خوارزميات التوجيه والتي تعمل في موجّهات الشبكة تتبادل وتحسب المعلومات التي تُستخدَم لتهيئة جداول التمرير تلك. وقد سبق توضيح التفاعل بين خوارزميات التوجيه وجداول التمرير في الشكل 4-2. بعد أن تعرفنا على بعض تفاصيل التمرير نحوّل انتباهنا الآن إلى الموضوع الأساسي الآخر في هذا الفصل والذي يمثل الوظيفة الهامة الأخرى لطبقة الشبكة، ألا وهو التوجيه. وسواء كانت طبقة الشبكة توفر خدمة وحدات البيانات (datagrams) (في هذه الحالة قد تأخذ الرزم مسارات مختلفة بين زوج ما من المُرسِل والمستقبل) أو خدمة الدائرة الافتراضية (VC) (في هذه الحالة ستتبع كل الرزم نفس المسار من المصدر إلى الوجهة)، فإنه يجب عليها أن تحدد المسارات التي

ستأخذها الرزم من المُرسِلين إلى المستقبلين. سنرى أن وظيفة التوجيه هي المسؤولة عن تحديد مسارات جيدة من المُرسِلين إلى المستقبلين خلال شبكة الموجّهات.

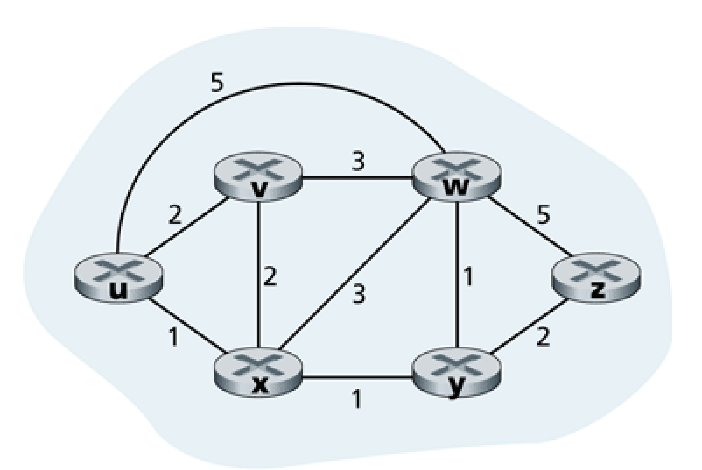
عادةً ما يوصّل كل مضيف مباشرة بموجّه واحد يمثل الموجّه الاعتيادي (default router) للمضيف (أيضاً يسمى موجّه القفزة الأولى للمضيف). وحينما يرسل مضيفٌ ما رزمة تنتقل الرزمة إلى الموجّه الاعتيادي له. نشير إلى الموجّه الاعتيادي لمضيف المصدر بموجّه المصدر والموجّه الاعتيادي لمضيف الوجهة بموجّه الوجهة. وبالتالي تصبح مشكلة توجيه رزمة من مضيف المصدر إلى مضيف الوجهة تماماً هي مشكلة توجيه الرزمة من موجّه المصدر إلى موجّه الوجهة، وهو ما سيمثل بؤرة التركيز لهذا الجزء.

إن الغرض من خوارزمية التوجيه بسيط: هو إيجاد مسار جيد من المصدر إلى الوجهة خلال مجموعة من الموجّهات المتصلة فيما بينها بوصلات. ويعرّف المسار الجيد بأنه أحد المسارات بين المصدر والوجهة والذي له أدنى كلفة. سنرى مع ذلك أنه من الناحية العملية تلعب بعض الاعتبارات الأخرى في الشبكات الحقيقية مثل قضايا سياسة الشبكة (مثلاً القاعدة "الموجّه x ينتمي للمنظمة y، ولذا يجب ألا ترسل إليه أيّة رزم مصدرها شبكة المنظمة z") دوراً أيضاً في تعقيد الخوارزميات البسيطة والرائعة التي تعتمد عليها شبكات اليوم.

يُستخدم رسم بياني (graph) لصياغة مشاكل التوجيه. تذكّر أن الرسم البياني $G = (N, E)$ هو مجموعة من العقد (النقط) N ومجموعة من الحافات (الخطوط) E حيث تمثّل كل حافة بزوج من العقد الموجودة في N. في سياق توجيه طبقة الشبكة تمثّل العقد في الرسم البياني الموجّهات (أي النقط التي يتم عندها اتخاذ قرارات التوجيه) وتمثل الحافات الوصلات المادية بين هذه الموجّهات. يوضح الشكل 4-27 مثالاً لهذا الرسم البياني المجرد لشبكة حاسب. يمكنك الاطلاع على بعض الرسوم البيانية التي تمثّل خرائط شبكات حقيقية في [Dodge 2007; Cheswick 2000]؛ وللاطلاع على مناقشة حول مدى تمثيل النماذج المختلفة المعتمدة

على الرسم البياني لشبكة الإنترنت انظر [Zegura 1997; Faloutsos 1999; Li 2004].

كما يوضّح الشكل 4-27، تقترن بكل حافة أيضاً قيمة تمثّل كلفتها، والتي قد تعكس الطول الطبيعي للوصلة المناظرة (على سبيل المثال قد يكون لوصلة عبر المحيطات كلفة أعلى من وصلة أرضية قصيرة المدى)، أو سرعة الوصلة، أو الكلفة النقدية للوصلة. لغرض التوجيه سنعتبر ببساطة أن لكل حافة كلفة دون الاكتراث بماهية تلك الكلفة. سنشير إلى كلفة أي حافة (x, y) في المجموعة E بالرمز $c(x, y)$ وهي تمثل كلفة الحافة بين العقدتين x وy. وإذا كانت الحافة (x, y) لا تنتمي إلى E تكون $c(x, y) = \infty$. أيضاً سنعتبر فقط الرسوم البيانية غير المتجهة (undirected graphs) (أي الرسوم البيانية ذات الحافات غير المتجهة)، وبالتالي تكون الحافة (x, y) هي نفسها تماماً الحافة (y, x)؛ وكذلك $c(x, y) = c(y, x)$. أيضاً يقال: إن العقدة y جار للعقدة x إذا كانت الحافة (x, y) تنتمي إلى E.



الشكل 4-27 نموذج رسم بياني تجريدي لشبكة حاسب.

بهذه الكلفة للحافات المختلفة في الرسم البياني التجريدي للشبكة يكون الهدف الطبيعي لخوارزمية التوجيه هو ايجاد المسارات الأقل كلفة بين المصادر والوجهات. ولجعل هذه المشكلة أكثر دقة تذكر أن المسار في الرسم البياني $G = (N, E)$ هو سلسلة من العقد $(x_1، x_2، ...، x_p)$ بحيث كل زوج من الأزواج $\{(x_1، x_2)، (x_2، x_3)، ....، (x_{p-1}، x_p)\}$ يمثل حافة ضمن E. تساوي كلفة المسار $(x_1، x_2، ...، x_p)$ مجموع كلف جميع الحافات على طول المسار أي

$$c(x_1, x_2) + c(x_2, x_3) + \ldots + c(x_{p-1}, x_p)$$

عادةً ما توجد عدة مسارات بين كل عقدتين x وy في الرسم البياني ولكلٍّ منها كلفة، ولواحد أو أكثر منها الكلفة الأدنى. يمكن الآن صياغة مسألة الكلفة الأدنى كالآتي: أوجد مساراً له أقل كلفة بين المصدر والوجهة. على سبيل المثال في الشكل 4-27 المسار أقل كلفة بين عقدة المصدر u وعقدة الوجهة w هو $(u، x، y، w)$ وله كلفة مقدارها 3. لاحظ أنه إذا كانت كل الحافات في الرسم البياني لها نفس الكلفة فسيكون المسار الأدنى كلفةً هو نفسه أقصر مسار (أي المسار الذي يتكون من أقل عدد من الوصلات بين المصدر والوجهة).

كتمرين بسيط حاول إيجاد مسار أقل كلفة من العقدة u إلى z في الشكل 4-27 وتأمل لبرهة كيف حسبت ذلك المسار. إذا كنت مثل الكثير من الناس ستحدد المسار المطلوب من u إلى z بفحص الشكل 4-27، مقتفياً بضعة مسارات من u إلى z وبطريقة ما ستقتنع نفسك أن المسار الذي اخترته هو الأقل كلفةً بين جميع المسارات المحتملة. هل فحصت كل المسارات المحتملة بين u وz (وعددها 17 مساراً في هذا المثال)؟ من المحتمل لا!. تُعتبر هذه العملية مثالاً لخوارزمية توجيه مركزية (centralized routing algorithm)، حيث يتم تشغيلها في موقع واحد (دماغك في هذا المثال) بمعلومات كاملة عن الشبكة. وبشكلٍ عام من الطرق التي يمكن بها تصنيف خوارزميات التوجيه هو كونها إما عالمية أو لامركزية:

- خوارزمية توجيه عالمية: تحسب المسار الأدنى كلفةً بين المصدر والوجهة بناءً على معرفة كاملة بجميع أجزاء الشبكة. أي أنها تستخدم الوصلات بين كل العقد وكلفة كل وصلة كمُدخلات. وهذا يتطلب في الحقيقة حصول

الخوارزمية على تلك المعلومات بطريقةٍ ما قبل إجراء حساب أفضل مسار. يمكن إجراء الحساب نفسه في موقع واحد (مركز خوارزمية التوجيه العالمية) أو في مواقع متعدّدة مكررة (replicated). ولكن الميزة الهامة لهذا النوع من الخوارزميات هي أنها تمتلك معلومات كاملة حول كل الوصلات وكلفتها. من الناحية العملية غالباً ما يشار إلى هذه الخوارزميات باسم خوارزميات حالة الوصلة ((Link State (LS) حيث يتعين أن تعرف الخوارزمية كلفة كل وصلة في الشبكة. سندرس خوارزميات LS في الجزء 4-5-1.

- خوارزمية توجيه لامركزية: في هذه الحالة يُنفّذ حساب المسار أقل كلفة بأسلوب موزّع تكراري. لا تمتلك أي عقدة المعلومات الكاملة عن كلفة كل الوصلات الموجودة بالشبكة، وإنما تبدأ كل عقدة بمعرفة كلفة الوصلات المتصلة بها مباشرة. ثم من خلال عملية تكرارية من حساب وتبادل المعلومات مع العقد المجاورة (أي العقد التي في النهايات الأخرى للوصلات المتصلة مباشرة بها)، وبشكلٍ تدريجي تحسب العقدة مسار أقل كلفة إلى وجهة ما أو إلى مجموعة من الوجهات. تدعى خوارزمية التوجيه اللامركزية التي سندرسها في الجزء 4-5-2 بخوارزمية متجه المسافة (DV)، لأن كل عقدة تحتفظ بمتجه لتقديرات الكلفة (أي المسافة) إلى كل العقد الأخرى في الشبكة.

كما يُمكن تصنيف خوارزميات التوجيه أيضاً حسب كونها ثابتة أو ديناميكية. في خوارزميات التوجيه الثابتة تتغيّر المسارات ببطء شديد بمرور الوقت، وغالباً كنتيجة للتدخّل البشري (كأن يعدّل شخصٌ ما يدوياً جدول التمرير بالموجّه). تُغيّر خوارزميات التوجيه الديناميكية المسارات مع تغيّر أحمال أو طبوغرافية الشبكة. يمكن أن تنفذ الخوارزمية الديناميكية إمّا بشكلٍ دوري أو كرد فعل مباشر للتغيرات في طبوغرافية الشبكة أو كلفة الوصلات. رغم أن الخوارزميات الديناميكية تمتاز باستجابتها السريعة للتغيرات في الشبكة، إلا أنها أيضاً تُعتبر أكثر عُرضةً للمشاكل مثل حلقات التوجيه (routing loops) (أي المسارات المغلقة) وتذبذب المسارات (route oscillation).

يعتمد أسلوب ثالث لتصنيف خوارزميات التوجيه على كونها تتأثر بأحمال الشبكة أو لا تتأثر. في الخوارزمية التي تتأثر بالأحمال، تتغيّر كلفة الوصلة ديناميكياً لتعكس المستوى الحالي للازدحام في تلك الوصلة. إذا تم تحديد كلفة عالية لوصلة مزدحمة حالياً، فإن خوارزمية التوجيه ستميل إلى اختيار مسارات بعيدة عن تلك الوصلة المزدحمة. وقد كانت خوارزميات التوجيه الأولى في شبكة Arpanet متأثرة بالأحمال [McQuillan 1980]، إلا أنها صادفت عدداً من الصعوبات [Huitema 1998]. خوارزميات التوجيه في الإنترنت اليوم (مثل RIP، OSPF، BGP) لا تتأثر بالأحمال لأن كلفة الوصلات لا تعكس بشكلٍ واضح مستوى الازدحام الحالي (أو في الماضي القريب).

4-5-1 خوارزمية التوجيه من نوع حالة الوصلة (LS)

تذكّر أنه في خوارزمية حالة الوصلة تكون طبوغرافية الشبكة وكلفة كل الوصلات معروفة (أي متوفرة كمدخلات إلى الخوارزمية). عملياً يتم ذلك بجعل كل عقدة تذيع رزماً عن حالة الوصلة إلى كل العقد الأخرى في الشبكة، حيث تحتوي كل رزمة على هويّات وكلف الوصلات الملحقة بها. عملياً (على سبيل المثال في بروتوكول التوجيه في الإنترنت OSPF والذي سنناقشه في الجزء 4-6-1) يتم ذلك في أغلب الأحيان بواسطة خوارزمية إذاعة حالة الوصلة [Perlrnan 1999]. سنغطّي خوارزميات الإذاعة في الجزء 4-7. ونتيجة لذلك ستحصل كل العقد على معلومات مماثلة عن الشبكة بكاملها. وعندئذ يمكن أن تُشغل كل عقدة خوارزمية LS عليها لحساب مجموعة المسارات ذات الكلفة الأدنى إلى العقد الأخرى.

سنقدم فيما يلي خوارزمية توجيه من نوع حالة الوصلة تُعرف بخوارزمية Dijkstra على اسم مخترعها. وهناك خوارزمية أخرى وثيقة الصلة بها يطلق عليها خوارزمية Prim؛ راجع [Cormen 2001] لمناقشة عامّة عن خوارزميات الرسم البياني. تحسب خوارزمية Dijkstra المسارات ذات الكلفة الأدنى من عقدة معينة (عقدة المصدر، وسنشير لها بـ u) إلى كل العقد الأخرى في الشبكة. خوارزمية Dijkstra تكرارية ولها خاصية أنه بعد التكرار k مرة تكون المسارات ذات الكلفة الأدنى

معروفة إلى عدد k من عقد الوجهات، وأنه من بين المسارات ذات الكلفة الأقل لجميع عقد الوجهات تكون تلك المسارات الـ k هي المسارات الـ k الأدنى كلفة.

دعنا نعرّف الرموز التالية:

- $D(v)$: كلفة المسار الأدنى كلفة من عقدة المصدر إلى الوجهة v خلال هذا التكرار للخوارزمية.
- $p(v)$: العقدة السابقة (v جار) على طول المسار الأدنى كلفة من المصدر إلى v.
- N': مجموعة جزئية من العقد؛ حيث تكون v ضمن N' إذا كان المسار الأدنى كلفة من المصدر إلى v معروفاً بشكلٍ حاسم.

تتكون خوارزمية التوجيه العالمية من خطوة تهيئة يتبعها حلقة تكرارية. عدد مرات تكرار الحلقة يساوي عدد العقد في الشبكة. عند الانتهاء تكون الخوارزمية قد حسبت أقصر مسارات من عقدة المصدر u إلى كل عقدة في الشبكة.

خوارزمية حالة الوصلة من عقدة المصدر u:

```
1  Initialization:
2     N' = {u}
3     for all nodes v
4        if v is a neighbor of u
5           then D(v) = c(u, v)
6        else  D(v) = ∞
7
8  Loop
9     find w not in N' such that D(w) is a minimum
10     add w to N'
11     update D(v) for each neighbor v of w and not in N':
12           D(v) = min{D(v), D(w) + c(w, v)}
13     /* new cost to v is either old cost to v or known
14           least path cost to w plus cost from w to v */
15 until N' = N
```

وكمثال دعنا نحسب المسارات الأدنى كلفة من u إلى جميع الوجهات المحتملة ضمن الشبكة المبينة في الشكل 4-27. يبين الجدول 4-3 ملخصاً بحسابات الخوارزمية بحيث يعطي كل سطر في الجدول قيم متغيرات الخوارزمية في نهاية كل تكرار. دعنا ننظر أولاً للخطوات القليلة الأولى بالتفصيل:

step	**N'**	**D(v), p(v)**	**D(w), p(w)**	**D(x), p(x)**	**D(y), p(y)**	**D(z), p(z)**
0	*u*	2,*u*	5,*u*	1,*u*	∞	∞
1	*ux*	2,*u*	4,*x*		2,*x*	∞
2	*uxy*	2,*u*	3,*y*			4,*y*
3	*uxyv*		3,*y*			4,*y*
4	*uxyvw*					4,*y*
5	*uxyvwz*					

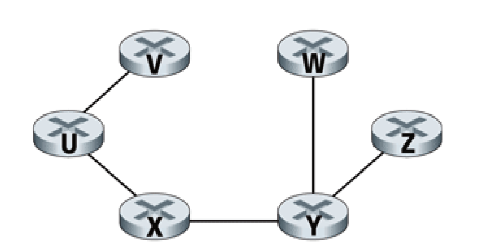
الجدول 4-3 تنفيذ خوارزمية حالة الوصلة على الشبكة الموجودة في الشكل 4-27.

- في خطوة التهيئة تحدد المسارات الأدنى كلفةً والمعروفة حالياً من عقدة المصدر إلى جيرانها الملحقين مباشرة v، x، w بالقيم 2، 1، 5 على التوالي. لاحظ بشكلٍ خاص أن الكلفة إلى w تأخذ القيمة 5 لأن هذه الكلفة هي الكلفة المباشرة (قفزة واحدة) للوصلة من u إلى w (رغم أننا سنرى قريباً أنه يوجد في الحقيقة مسار بكلفة أقل). أما الكلف إلى y وz فتأخذ القيمة لانهاية لعدم وجود وصلة مباشرة بين كلٍّ منهما والمصدر.
- في التكرار الأول نبحث بين العقد التي لم تضاف بعد إلى المجموعة N' عن العقدة التي لها أدنى كلفة من المصدر في نهاية التكرار السابق. تلك العقدة هي x وبكلفة تساوي 1، وهكذا تضاف x إلى المجموعة N'. يُنفّذ السطر 12 من خوارزمية LS حينئذ لتحديث قيمة $D(v)$ لكل العقد v فنحصل على النتائج المبينة في السطر الثاني (الخطوة 1) في الجدول 4-3. لا تتغيّر كلفة المسار إلى v. أما كلفة المسار إلى w (والتي كانت 5 في نهاية التهيئة) عبر العقدة x فلها الآن كلفة تساوي 4. لذا نختار هذا المسار الأدنى كلفة ونعدل

العقدة السابقة لـ w على طول المسار الأقصر من u لتصبح x. بنفس الطريقة تكون الكلفة إلى y (عبر x) تساوي 2، ويعدّل الجدول وفقاً لذلك.

- في التكرار الثاني نجد أن العقد v وy لها أدنى كلفة (تساوي 2)، لذا سنختار أحدهما بشكلٍ اعتباطي وليكن y ونضيفها إلى المجموعة N' والتي ستتضمن الآن u، x، y. ثم نعدل الكلفة إلى العقد الأخرى غير الموجودة في N' (أي العقد v، w، z) عن طريق السطر 12 في خوارزمية LS فنحصل على الناتج الموضح في السطر الثالث في الجدول 4-3.
- وهكذا .....

عندما تنتهي خوارزمية LS نكون قد حصلنا لكل عقدة على سلفها (predecessor) على طول مسار أدنى كلفة من عقدة المصدر إلى تلك العقدة، ولكل سلف لدينا سلفه أيضاً وهكذا يمكننا أن نبني كامل المسار من المصدر إلى كل الوجهات. ومن ثم يمكن أن نبني جدول التمرير في كل عقدة (مثلاً العقدة u) وذلك بتخزين لكل وجهة عقدة القفزة التالية على مسار أدنى كلفة من u إلى الوجهة. يبين الشكل 4-28 مسارات أدنى كلفة وجدول التمرير في العقدة u للشبكة الموجودة في الشكل 4-27.



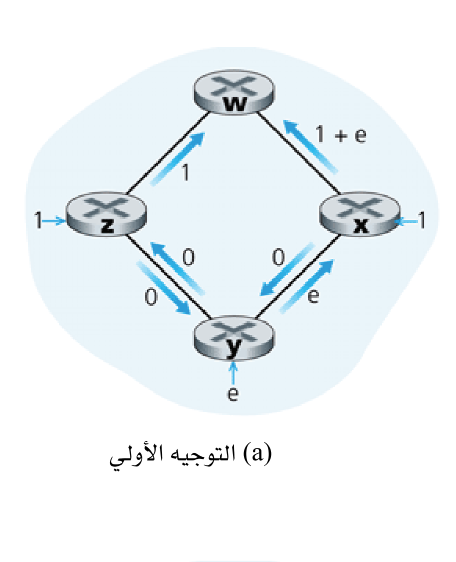
وصلة	واجهة
(u, v)	v
(u, x)	w
(u, x)	x
(u, x)	y
(u, x)	z

الشكل 4-28 المسارات الأدنى كلفة وجدول التمرير على العقدة u.

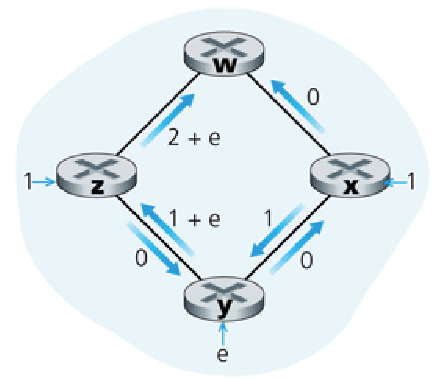
ما حجم التعقيد الحسابي (computational complexity) لهذه الخوارزمية؟ أي ما كمية الحسابات التي يجب إنجازها في أسوأ الأحوال لإيجاد المسارات الأدنى كلفة من المصدر إلى وجهات عددها n عقدة؟ في التكرار الأول نحتاج للبحث خلال كل العقد n لتقرير العقدة w غير الموجودة في N' ولها أدنى كلفة. في التكرار الثاني نحتاج للبحث في n -1 عقدة لتقرير الكلفة الدنيا، وفي التكرار الثالث n - 2 عقدة، وهكذا. وبالتالي يكون العدد الكلي للعقد التي نحتاج البحث خلالها في كل التكرارات يساوي $n(n + 1)/2$، وهكذا نقول بأن التطبيق السابق لخوارزمية LS له تعقيد في أسوأ الأحوال $O(n^2)$. وهناك تطبيق أكثر تطوّراً لهذه الخوارزمية باستعمال هياكل بيانات تُعرف بالكومة (heap)، حيث يمكن إيجاد الحد الأدنى في السطر 9 في زمن لوغاريتمي بدلاً من زمن خطي مما يقلل التعقيد.

قبل أن نغادر خوارزمية LS، دعنا نناقش بعض المشاكل التي يمكن أن تظهر. يبين الشكل 4-29 طبوغرافية شبكة بسيطة وفيها كلفة الوصلة تساوي الحمل الفعلي للوصلة، ومن ثم يعكس على سبيل المثال التأخير الذي سيواجَه. في هذا المثال كُلَف الوصلات ليست متماثلة؛ أي أن $c(u, v)$ تساوي $c(v, u)$ فقط إذا كان الحمل الفعلي في كلا الاتجاهين على الوصلة (u, v) متساوياً. في هذا المثال افترض أن العقدة z تُرسل وحدة بيانات إلى w، والعقدة x أيضا تُنشئ وحدة بيانات إلى w، أما العقدة v فترسل كمية بيانات مقدرها e إلى w أيضاً. يبين الشكل 4-29 (a) التوجيه الأولي حيث تناظر كلفة الوصلة كمية البيانات التي تنقل خلالها.

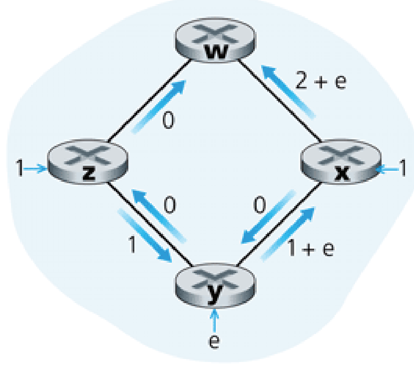
عند تشغيل خوارزمية LS في المرة التالية تقرر العقدة y (بناءً على كلفة الوصلة في الشكل 4-29 (a)) أن المسار في اتجاه عقارب الساعة إلى w له كلفة 1، بينما المسار في عكس عقارب الساعة (والذي تستخدمه حالياً) له كلفة تساوي e+1. لذلك تُعدِّل الآن مسارها إلى w باتجاه عقارب الساعة. وبنفس الطريقة تقرّر x أن المسار الجديد أقل كلفة إلى w في اتجاه عقارب الساعة أيضاً، وبالتالي نحصل على الكُلَف في الشكل 4-29 (b). عند تشغيل خوارزمية LS في المرة التالية تكتشف العقد x، y، z وجود مسار بكلفة صفر إلى w في اتجاه عكس عقارب الساعة، وبالتالي تُغيِّر مساراتها في اتجاه عكس عقارب الساعة. في المرة التالية لتشغيل خوارزمية LS توجه x، y، z مساراتها باتجاه عقارب الساعة، وهكذا.



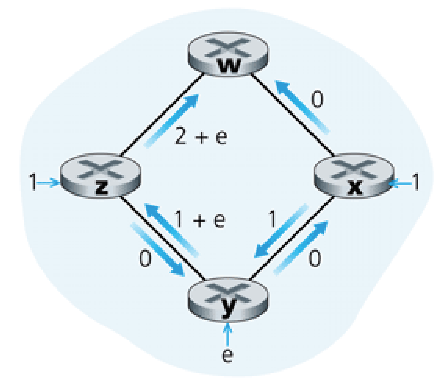
(a) التوجيه الأولي



(b) تكتشف x، y مساراً أفضل إلى w في اتجاه عقارب الساعة



(c) تكتشف x، y، z مساراً أفضل إلى w في اتجاه عكس عقارب الساعة



(d) تكتشف x، y، z مساراً أفضل إلى w في اتجاه عقارب الساعة

الشكل 4-29 تذبذبات في التوجيه الحساس للازدحام.

ماذا يُمكن فعله لمنع مثل تلك التذبذبات (والتي يمكن أن تحدث في أي خوارزمية تستخدم معياراً يعتمد على الازدحام أو التأخير على الوصلة كخوارزمية LS)؟ يتطلب أحد الحلول ألا تعتمد كلف الوصلات على كمية المرور التي تحملها الوصلة فعلاً، وهو حل غير مقبول لأنه قد يكون من أهداف التوجيه تجنب مشاكل الازدحام على الوصلات (كالتأخير العالي مثلاً). يكمن حل آخر في

تجنب تشغيل خوارزمية LS في كل الموجّهات في نفس الوقت. هذا الحل يبدو أكثر معقولية، حيث إننا نأمل أنه رغم استخدام الموجّهات لخوارزمية LS بنفس الدورية (periodicity) فإن لحظة التنفيذ لن تكون نفسها في كل عقدة. وبشكلٍ مثير للانتباه وجد الباحثون أنه يمكن أن يحدث تزامن ذاتي فيما بين موجّهات الإنترنت [Floyd Synchronization 1994]. بمعنى أنه رغم أن الموجّهات تنفّذ في البداية الخوارزمية بنفس الفترة الدورية وفي لحظات مختلفة لكن يمكن أن يؤول الأمر في النهاية إلى أن تنفذ الموجّهات الخوارزمية بصورة متزامنة (ومن ثم تستمر على هذا الحال). أحد طرق تفادي مثل هذا التزامن الذاتي هو أن يختار كل موجّه وقت الإعلان عن حالة الوصلة بطريقة عشوائية.

بعد أن تناولنا خوارزمية LS دعنا نناقش خوارزمية التوجيه الرئيسة الأخرى (أي خوارزمية توجيه متجه المسافة).

4-5-2 خوارزمية توجيه متجه المسافة (DV)

بينما تستخدم خوارزمية LS معلومات عالمية، تُعتبر خوارزمية توجيه متجه المسافة (DV) موزّعة وتكرارية ولاتزامنية. فهي موزّعة حيث إن كل عقدة تستلم بعض المعلومات من واحد أو أكثر من جيرانها الموصّلين بها مباشرة، وتقوم بإجراء الحسابات، ثم بعد ذلك توزّع النتائج إلى جيرانها. وهي تكرارية حيث إن هذه العملية تستمر حتى ينتهي تبادل المزيد من المعلومات بين الجيران. (وبشكلٍ هام هذه الخوارزمية ذاتية الانتهاء أيضاً حيث لا توجد إشارة لإنهاء الحسابات وإنما فقط تنتهي). وأخيراً هي خوارزمية لاتزامنية حيث إنها لا تتطلّب أن تعمل كل العقد بإيقاع موحد مع بعضها البعض. سنرى بعد قليل كيف أن خوارزمية لاتزامنية وتكرارية وذاتية الانتهاء وموزّعة تكون أكثر تشويقاً ومتعة من خوارزمية مركزية!

قبل أن نقدّم خوارزمية DV من المفيد مناقشة علاقة مهمة توجد بين كُلَف مسارات أدنى كلفة. لنرمز لكلفة مسار أدنى كلفة من العقدة x إلى العقدة y بـ

$d_x(y)$. عندئذ ترتبط الكلف بمعادلة بلمن - فورد (Bellman-Ford) الشهيرة كما يلي:

$$d_x(y) = \min_v\{c(x, v) + d_v(y)\} \quad (4\text{-}1)$$

حيث إن $\min_v$ في المعادلة تطبق على كل جيران x. إن معادلة بلمن - فورد بدهية نوعاً ما. في الحقيقة بعد الانتقال من x إلى v نأخذ عندئذ المسار الأدنى كلفة من v إلى y، وبالتالي تكون كلفة المسار $c(x,v) + d_v(y)$. ولأننا يجب أن نبدأ بالانتقال إلى أحد الجيران v، فإن أدنى كلفة من x إلى y تكون أصغر قيمة لـ $c(x,v) + d_v(y)$ على كل الجيران.

لكن لأولئك الذين قد يساورهم الشك في صحة المعادلة، دعنا نختبرها لعقدة المصدر u وعقدة الوجهة z في الشكل 4-27. لعقدة المصدر u ثلاثة جيران: v، x، w. بتتبع المسارات المختلفة في الرسم البياني من السّهل رؤية أن $d_v(z)=5$، $d_x(z)=3$، $d_w(z)=3$. بتعويض هذه القيم في المعادلة 4-1 مع الكلف $c(u,v)=2$، $c(u,x)=1$، $c(u,w)=5$ نحصل على $d_u(z) = \min\{2+5, 5+3, 1+3\} = 4$. من الواضح أن هذا صحيح ويمثل تماماً ما نحصل عليه من خوارزمية Dijskstra لنفس الشبكة. هذا التحقّق السريع يجب أن يساعد في التخفيف من أي تشكك عندك.

إن معادلة بلمن - فورد ليست مجرد فضول ثقافي، وإنما لها في الحقيقة أهمية عملية هامّة. وبشكلٍ محدد يعطي الحل لمعادلة بلمن - فورد المدخلات في جدول التمرير للعقدة x. ولرؤية هذا نرمز بـ v^* لأي عقدة مجاورة لها الحدّ الأدنى في المعادلة 4-1. عندئذ إذا أرادت العقدة x إرسال رزمة إلى العقدة y على المسار الأدنى كلفة، يجب أن ترسل الرزمة أولاً إلى العقدة v^*. وهكذا يحتوي جدول التمرير للعقدة x العقدة v^* كموجّه القفزة التالية للوجهة النهائية y. وكمساهمة عملية مهمة أخرى لمعادلة بلمن - فورد فإنها تقترح شكل الاتصال من جار لجار والذي يحدث في خوارزمية DV.

تتلخص الفكرة الأساسية في التّالي: تبدأ كل عقدة x بتقدير $D_x(y)$ لكلفة مسار أدنى كلفة منها إلى العقدة y، وذلك لكل العقد في N. ولنرمز لمتجه المسافة

للعقدة x بـ $\boldsymbol{D}_x = [D_x(y): y \in N]$. تحتفظ كل عقدة x في خوارزمية DV بمعلومات التوجيه التالية:

- الكلفة $c(x,v)$ من x لكل جار v موصّل بها مباشرة.
- متجه المسافة للعقدة x أي $\boldsymbol{D}_x = [D_x(y): y \in N]$ والذي يحتوي على مسافات تقديرية لكل وجهة y في N.
- متجه المسافة لكل عقدة من جيرانها أي $\boldsymbol{D}_v = [D_v(y): y \in N]$ لكل جار من جيران العقدة x.

من وقت لآخر في الخوارزمية اللاتزامنية الموزّعة ترسل كل عقدة نسخة من متجه المسافة الخاص بها إلى كل جيرانها. عندما تستلم العقدة x متجه مسافة جديد من أيٍّ من جيرانها وليكن v تقوم بتخزين متجه المسافة للعقدة v ثم بعد ذلك تستخدم معادلة بلمن - فورد لتحديث متجه المسافة الخاص بها كالتّالي:

$$D_x(y) = \min_v\{c(x, v)+D_v(y)\}$$

لجميع العقد y في N. إذا تغيّر متجه المسافة للعقدة x كنتيجة لخطوة التحديث هذه، عندئذ سترسل العقدة x متجه المسافة الجديد الخاص بها إلى كل جيرانها، والذين يمكن تباعاً أن يُعدِّلوا متجهات المسافة الخاصة بهم. وطالما أن كل العقد تواصل تبادل متجهات المسافة بشكلٍ غير متزامن، فإن تقديرات المسافة $D_x(y)$ ستؤول للقيمة الحقيقية لمسار أدنى كلفة من x لـ y أي $d_x(y)$ [Bertsekas 1991].

في خوارزمية DV تقوم العقدة x بتحديث متجه المسافة التقديري عندما ترى تغيّراً في كلفة إحدى وصلاتها المرتبطة بها مباشرة أو عندما تستلم متجه مسافة جديد من أحد جيرانها. لكن لتحديث جدول التمرير الخاص لوجهة معينة y تحتاج العقدة x حقاً لمعرفة ليس فقط مسافة المسار الأقصر إلى y وإنما العقدة المجاورة $v^*(y)$ التي تمثل أول قفزة على طول المسار الأقصر إلى y. وربما تكون قد توقّعتَ أن موجّه القفزة التالية $v^*(y)$ هو العقدة المجاورة التي تحقق أصغر قيمة في السطر 14 لخوارزمية DV. إذا كان هناك عدة جيران v لهم نفس القيمة الصغرى فيمكن استخدام أيٍّ منهم. هكذا في السطور 13-14 تقوم العقدة x أيضاً بتحديد $v^*(y)$ وتحديث جدول التمرير لكل وجهة y.

خوارزمية متجه المسافة (DV)

في كل عقدة x:

```
1  Initialization:
2    for all destinations y in N:
3      Dx(y)=c(x, y)    /* if y is not a neighbor then c(x, y) = ∞ */
4    for each neighbor w
5      Dw(y)= ∞ for all destinations y in N
6    for each neighbor w
7      send distance vector Dx = [Dx(y): y in N] to w
8
9  Loop
10   wait (until I see a link cost change to some neighbor w or
11        until I receive a distance vector from some neighbor w)
12
13   for each y in N:
14      Dx(y) = minv{c(x, v) + Dv(y)}
15
16   if Dx(y) changes for any destination y
17        send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19  forever
```

تذكّر أن خوارزمية LS عالمية بمعنى أنها تتطلّب من كل عقدة الحصول أولاً على خريطة كاملة للشبكة قبل تشغيل خوارزمية Dijkstra؛ في حين خوارزمية DV لامركزية ولا تستخدم مثل هذه المعلومات العالمية. في الحقيقة المعلومات الوحيدة التي لدى كل عقدة هي كلفة كل الوصلات إلى جيرانها الملحقين بها مباشرةً والمعلومات التي تستقبلها من هؤلاء الجيران. تنتظر كل عقدة رسالة تحديث من أي جار (السطور 10-11)، وتحسب متجه المسافة الجديد عندما تستقبل رسالة التحديث تلك (السطر 14)، ثم توزّع متجه المسافة الجديد إلى جيرانها (السطور 16-17). تستخدم الخوارزميات التي تشبه DV عملياً في العديد من بروتوكولات

التوجيه، بما في ذلك بروتوكولات الإنترنت RIP وBGP، وبروتوكول ISO IDRP، وبروتوكول Novell IPX، وبروتوكول شبكة Arpanet الأصلي.

يوضح الشكل 4-30 طريقة عمل خوارزمية DV لشبكة بسيطة مكونة من ثلاث عقد والموضحة في أعلى الشكل. إن طريقة عمل الخوارزمية مصوّرة بطريقة متزامنة، حيث تستلم كل العقد متجهات المسافة بشكلٍ آني من جيرانها، وتحسب متجهات المسافة الجديدة الخاصة بها، ومن ثمَّ تُخبر جيرانها إذا تغيّرت متجهات المسافة. يجب أن تقنع نفسك بعد هذا المثال أنه يمكن إجراء الخوارزمية بشكلٍ صحيح في أسلوب لاتزامني أيضاً، بمعنى إجراء الحسابات وإرسال واستقبال رسائل التحديث في أي وقت كان.

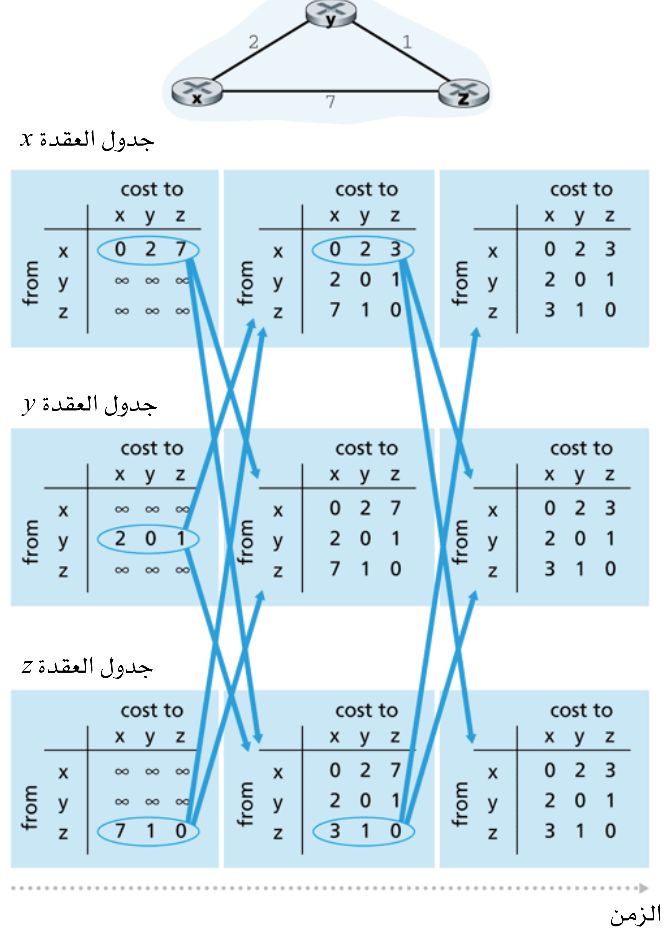
يعرض العمود في أقصى اليسار للشكل ثلاثة جداول توجيه أوّليَّة لكلٍّ من العقد الثلاث. على سبيل المثال الجدول في الزاوية اليسرى من أعلى هو جدول التوجيه الأوَّلي للعقدة x. بداخل جدول توجيه معين يمثل كل صف متجه مسافة. وبالتحديد يتضمّن جدول التوجيه لكل عقدة متجه المسافة الخاص بها وذلك الخاص بكلٍّ من جيرانها. وهكذا يكون الصف الأول في جدول التوجيه الأوَّلي للعقدة x كالآتي:

$$\boldsymbol{D}_x=[D_x(x), D_x(y), D_x(z)]= [0, 2, 7]$$

أما الصفان الثاني والثالث في هذا الجدول فيمثلان متجهي المسافة المستلمين مؤخراً من العقد y وz على التوالي. ولأنه عند التهيئة لم تستلم العقدة x أي شيء من العقد y وz فإن المدخلات في الصفين الثاني والثالث تأخذ القيمة ما لانهاية.

بعد التهيئة ترسل كل عقدة متجه المسافة الخاص بها إلى كلٍّ من جيرانها. هذا موضح في الشكل 4-30 بالأسهم من جداول العمود الأول إلى جداول العمود الثاني. فمثلاً ترسل العقدة x متجه المسافة $\boldsymbol{D}_x$=[0, 2, 7] إلى كلٍّ من y وz. بعد استلام رسالة التحديث تلك تقوم كل عقدة بإعادة حساب متجه المسافة الخاص بها. على سبيل المثال تحسب العقدة x:

$$D_x(x) = 0$$
$$D_x(y)=\min\{c(x,y)+D_y(y), c(x,z)+D_z(y)\}=\min\{2+0, 7+1\}=2$$
$$D_x(z)=\min\{c(x,y)+D_y(z), c(x,z)+D_z(z)\}=\min\{2+1, 7+0\}=3$$



الشكل 4-30 خوارزمية متجه المسافة (DV).

لذا يعرض العمود الثاني لكل عقدة متجه المسافة الجديد للعقدة سويةً مع متجهات المسافة التي استلمتها للتّو من جيرانها. لاحظ أن تقديرات العقدة x – على سبيل المثال – لأدنى كلفة إلى العقدة z أي $D_x(z)$ قد تغيّرت من 7 إلى 3. لاحظ أيضاً أن العقدة y كأحد جيران العقدة x تحقق لها أدنى قيمة في السطر 14 من

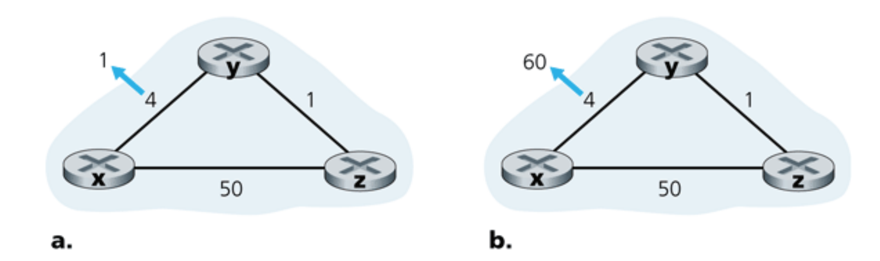
خوارزمية DV؛ وبهذا تصبح y عقدة أول قفزة للعقدة x في هذه المرحلة $v^*(y)=y$, $v^*(z)=y$.

بعد انتهاء العقد من حساب متجهات المسافة الخاصة بها تقوم بإرسالها مجدداً إلى جيرانها (في حالة حدوث تغيير)، كما هو موضح في الشكل 4-30 بالأسهم من جداول العمود الثاني إلى جداول العمود الثالث. لاحظ أن العقد x وz فقط هي الوحيدة التي ترسل رسائل تحديث؛ أما متجه المسافة للعقدة y فلم يتغيّر ولذا لا ترسل العقدة y رسالة تحديث. بعد استلام العقد لرسائل التحديث تقوم بإعادة حساب متجهات المسافة وتحديث جداول التوجيه كما هو موضح في العمود الثالث.

تتكرر عملية استلام متجهات المسافة من الجيران، وإعادة حساب مدخلات جدول التوجيه، وإعلام الجيران بالتغيّرات في كلف مسارات أدنى كلفة إلى الوجهة إلى أن يتوقف إرسال رسائل تحديث جديدة. عند هذه النقطة ونظراً لعدم استقبال رسائل تحديث لا تتم أية تعديلات أخرى على جداول التوجيه وتدخل الخوارزمية حالة خمود (أي تنفذ كل العقد الانتظار في السطور 10-11 من خوارزمية DV). تبقى الخوارزمية في حالة الخمود حتى تتغيّر كلفة وصلة كما سنناقش فيما يلي.

خوارزمية متجه المسافة: تغيّر كلفة وصلة أوعطلها

عندما تكتشف عقدة تستخدم خوارزمية DV تغيّراً في كلفة وصلة بينها وبين أحد جيرانها (السطور 10-11)، تقوم بتحديث متجه المسافة الخاص بها (السطور 13-14) وإذا حدث تغيّر في كلفة مسار أدنى كلفة فإنها تخبر جيرانها (السطور 16-17) بمتجه المسافة الجديد الخاص بها. يوضح الشكل 4-31 (a) سيناريو تغيّرت فيه كلفة الوصلة بين y وx من 4 إلى 1. سنركّز هنا فقط على مدخلات جداول العقد y وz إلى الوجهة x. تتسبّب خوارزمية DV في سلسلة الأحداث التالية:



الشكل 4-31 تغيّر كلفة الوصلة.

- في الوقت t_0 تكتشف y تغيّر كلفة الوصلة (من 4 إلى 1) فتقوم بتحديث متجه المسافة الخاص بها، وتخبر جيرانها بهذا التغيير.
- في الوقت t_1 تستلم z رسالة التحديث من y فتقوم بتحديث جدولها. تحسب z المسار الجديد الأدنى كلفة إلى x (تقل الكلفة من 5 إلى 2) وترسل متجه المسافة الجديد إلى جيرانها.
- في الوقت t_2 تتسلم y رسالة تحديث من z وتعدّل متجه المسافة الخاص بها، غير أن أقل كلفة لا تتغيّر ولذا لا ترسل أي رسالة إلى z. عندها تدخل الخوارزمية حالة خمود.

وهكذا تحتاج خوارزمية DV فقط لتكرارين حتى تصل لحالة الخمود. تنتشر الأخبار الجيدة حول النقص في الكلفة بين x وy بسرعة خلال الشبكة.

دعنا نرى الآن ما يمكن أن يحدث عندما تزيد كلفة وصلة. افترض بأن كلفة الوصلة بين x وy زادت من 4 إلى 60 كما هو موضح في الشكل 4-31 (b).

1. قبل أن تتغيّر كلفة الوصلة $D_y(x) = 4$، $D_y(z) = 1$، $D_z(y) = 1$، $D_z(x) = 5$. في الوقت t_0 تكتشف y تغيّر كلفة الوصلة (تغيّرت الكلفة من 4 إلى 60). تحسب y المسار الجديد الأدنى كلفة إلى x لتصبح

$$D_y(x)=\min\{c(y,x)+D_x(x),\ c(y,z)+D_z(x)\}=\min\{60+0,\ 1+5\}=6$$

بالطبع مع نظرتنا الكلية للشبكة يمكن أن نرى أن هذه الكلفة الجديدة عن طريق z خاطئة. لكن المعلومات الوحيدة المتوفرة لدى العقدة y هي أن الكلفة المباشرة إلى x تساوي 60 وأن z قد أخبرت y أخيراً بأنها يمكنها

الوصول إلى x بكلفة 5. لذا لكي تصل y إلى x ستوجّه الآن عبر z وتتوقّع أن تكون z قادرة على الوصول إلى x بكلفة 5.
عند الوقت t_1 تتكون لدينا حلقة توجيه مفرغة (مسار مغلق): فلكي تصل y إلى x توجّه إلى z، ولكن z توجه إلى x عبر y. إن حلقة التوجيه تشبه ثقباً مظلماً (black hole)، حيث تبقى الرزمة الموجهة إلى x ترتد من y إلى z والعكس ابتداءً من زمن t_1 إلى الأبد (أو إلى أن تتغيّر جداول التمرير).

2. لأن العقدة y قد حسبت حداً أدنى جديداً للكلفة إلى x، فإنها تخبر z بمتجه المسافة الجديد عند الزمن t_1.
3. في وقتٍ ما بعد t_1 تتسلم z متجه المسافة الجديد الخاص بـ y، والذي يشير بأن أقل كلفة من y إلى x تساوي 6. عندها تعلم z بأنه يمكن أن تصل إلى y بكلفة مقدرها 1 ومن ثمَّ تحسب أقل كلفة جديدة إلى x من $D_z(x)=\min\{50+0, 1+6\}=7$. ونظراً لزيادة أدنى كلفة من z إلى x فإنها تخبر y بمتجه المسافة الجديد عند t_2.
4. بطريقة مماثلة بعد استلام متجه المسافة الجديد الخاص بـ z تحدد y أن $D_y(x)=8$ وترسل إلى z متجه المسافة الخاص بها. ومن ثمَّ تقرّر z بأن $D_z(x)=9$ وترسل إلى y متجه المسافة الخاص بها، وهكذا.

إلى متى ستستمر تلك العملية؟ يجب أن تقنع نفسك أن الحلقة تُكرَّر 44 مرة (تبادل رسائل بين y وz) حتى تحسب z في النهاية كلفة مسارها عن طريق y ليصبح أكبر من 50. في هذه النقطة ستقرر z (أخيراً!) أن مسار أدنى كلفة إلى x هو عن طريق الوصلة المباشرة إلى x. عندئذ ستوجه y إلى x عن طريق z. النتيجة هي أن الأخبار السيئة حول الزيادة في كلفة الوصلة تنتشر في الحقيقة ببطء! ماذا كان سيحدث إذا كانت كلفة الوصلة $c(y, x)$ قد تغيّرت من 4 إلى 10000 وكانت الكلفة $c(z, x) = 9999$؟ بسبب مثل هذه السيناريوهات يطلق على هذه المشكلة أحياناً اسم مشكلة العد لما لانهاية (count-to-infinity problem).

خوارزمية متجه المسافة: إضافة اتجاه عكسي مسمّم (Poisoned Reverse)

يمكن تجنب سيناريو حلقة التوجيه المفرغة الذي تم وصفه للتو باستخدام تقنية تعرف بالاتجاه العكسي المسمّم. إن الفكرة بسيطة؛ إذا وجهت z من خلال y للوصول إلى الوجهة x فإن z تُعلِم y أن المسافة منها إلى x مالانهاية، أي أن $D_z(x) = \infty$ رغم معرفتها حالياً أنه في الحقيقة $D_z(x)=5$. تستمر z في إطلاق هذه الكذبة البيضاء الصغيرة إلى y طالما توجّه z إلى x عن طريق y. ولأن y تعتقد أن z ليس لديها مسار إلى x فلن تحاول y التوجيه إلى x عن طريق z طالما أن z تواصل التوجيه إلى x عن طريق y (وتستمر في كذبها حول ذلك!).

دعنا الآن نرى كيف أن تسميم الاتجاه العكسي يحلّ مشكلة حلقة التوجيه المفرغة التي واجهناها من قبل في الشكل 4-31 (b). كنتيجة لتسميم الاتجاه العكسي يشير جدول المسافة الخاص بـ y لأن $D_z(x) = \infty$. عندما تتغيّر كلفة الوصلة (x, y) من 4 إلى 60 في الوقت t_0 تقوم y بتحديث جدولها وتستمر في التوجيه مباشرة إلى x رغم أن الكلفة الآن أعلى وتساوي 60، وتخبر z الكلفة الجديدة إلى x، (أي أن $D_y(x)=60$). بعد استلام z لرسالة التحديث في الوقت t_1 تحوّل طريقها فوراً إلى x ليكون عن طريق الوصلة (z, x) وبكلفة تساوي 50. ولأن هذا يمثل مساراً جديداً إلى x ولأن المسار لم يعد يمرّ من خلال y، سوف تخبر z الآن y بأن $D_z(x)=50$ في الوقت t_2. بعد استلام رسالة التحديث من z تقوم y بتحديث جدول المسافة ليتضمن $D_y(x)=51$. أيضاً لأن z الآن على مسار أقلّ كلفة من y إلى x تقوم y بتسميم المسار العكسي من z إلى x بإعلام z في الوقت t_3 أن $D_y(x)=\infty$ (رغم أن y في الحقيقة تعرف أن $D_y(x)=51$).

هل تسميم الاتجاه العكسي يحل أيضاً مشكلة العد لما لانهاية؟ الجواب لا. يجب أن تقنع نفسك بأن الحلقات التي تتضمّن ثلاثة عقد أو أكثر لن يتم اكتشافها بأسلوب تسميم الاتجاه العكسي.

مقارنة بين خوارزميات التوجيه LS وDV

تتبع خوارزميات DV وLS طرقاً تتكامل فيما بينها لحساب التوجيه. في خوارزمية DV تخاطب كل عقدة جيرانها المرتبطين بها مباشرة فقط، لكنها تزودهم بتقديرات أدنى كلفة للمسارات بينها وبين كل العقد (التي تعرفها) في الشبكة. في خوارزمية LS تخاطب كل عقدة جميع العقد الأخرى (عن طريق الإرسال الإذاعي)، لكنها تخبرهم بكلف الوصلات المرتبطة مباشرة بها فقط. دعنا ننهي دراستنا لخوارزميات LS وDV بمقارنة سريعة لبعض خواصهما. تذكّر أن N هي مجموعة العقد (الموجّهات) وE هي مجموعة الحافات (الوصلات).

- تعقيد الرسالة (message complexity): لقد رأينا أن خوارزمية LS تحتاج لمعرفة كل عقدة بكلفة كل وصلة في الشبكة، مما يتطلب أن تكون الرسائل المرسلة $O(|N|.|E|)$. أيضاً عندما تتغيّر كلفة وصلة يجب أن ترسل الكلفة الجديدة للوصلة إلى كل العقد. أما خوارزمية DV فتتطلب تبادل الرسائل بين الجيران المرتبطين مباشرة فقط في كل تكرار. كما رأينا أن الوقت الذي تستغرقه الخوارزمية للتقارب يمكن أن يعتمد على العديد من العوامل. وعند تغيير كلفة الوصلة فإن خوارزمية DV ستبث نتائج الوصلة التي تغيرت كلفتها فقط إذا أدت كلفة الوصلة الجديدة إلى مسار جديد أقل كلفةً لأحد العقد المرتبطة بتلك الوصلة.
- سرعة التقارب (speed of convergence): لقد رأينا أن تحقيق خوارزمية LS من الدرجة $O(|N|^2)$ ويتطلب رسائل $O(|N|.|E|)$. يمكن أن تتقارب خوارزمية DV ببطء ويمكن أن تعاني من حلقات التوجيه المفرغة أثناء تقارب الخوارزمية. تعاني DV أيضاً من مشكلة العد لما لانهاية.
- المتانة (robustness): ماذا يمكن أن يحدث لو تعطل أحد الموجّهات، أو أساء التصرف، أو تم اختراقه؟ في حالة LS يمكن أن يذيع الموجّه كلفة خاطئة لأحد وصلاته المرتبطة به مباشرة (لكن يقتصر الخطأ على هذا التصرف). يمكن أيضاً أن تخرّب أحد العقد أو تسقط أياً من الرزم التي تستلمها كجزء من إرسال إذاعة ضمن خوارزمية LS. لكن عقدة LS تحسب جداول

التمرير الخاصة بها فقط، وتقوم العقد الأخرى بحسابات مماثلة لنفسها. هذا يعني فصل حسابات المسارات بعض الشيء عند استخدام LS، مما يزيد درجة متانة الخوارزمية. عند استخدام DV يمكن أن تعلن عقدة مسارات أدنى كلفةً خاطئة لأي وجهة من الوجهات أو لها جميعاً. مثال ذلك ما حدث عام 1997 حيث أدى عطب بموجّه في شبكة موفر خدمة إنترنت صغير إلى تزويد موجّهات شبكة العمود الفقري القومية بمعلومات توجيه خاطئة. نتج عن ذلك إغراق موجّهات أخرى للموجّه المتعطل بحركة مرور البيانات وبقيت أجزاء كبيرة من الإنترنت مقطوعة لعدّة ساعات [Neumann 1997]. وبتعميم أكثر نلاحظ أنه في كل تكرار يُنقل ناتج حساب عقدة في DV إلى جيرانها وبعد ذلك بشكلٍ غير مباشر إلى جيران جيرانها في التكرار التالي. وهذا يعني أنه يمكن أن ينتشر حساب عقدة خاطئ خلال الشبكة بكاملها عند استخدام DV.

في النهاية يتضح أن أياً من خوارزميات LS وDV لا يُعتبر فائزاً على الآخر، فكلاهما مستعمل في الإنترنت في حقيقة الأمر.

خوارزميات التوجيه الأخرى

لا تُعتبر خوارزميات LS وDV التي درسناها فقط واسعة الانتشار عملياً، ولكنهما بالضرورة بمثابة خوارزميات التوجيه الوحيدة المستعملة واقعياً اليوم في الإنترنت. ومع ذلك فقد تم اقتراح خوارزميات توجيه أخرى عديدة من قبل الباحثين خلال السنوات الـ 30 الماضية، تتراوح من البسيط جداً إلى المتطوّر والمعقّد جداً. يعتمد أحد الأنواع العامة لخوارزميات التوجيه على النظر إلى مرور الرزم كتدفقات (flows) بين المصادر والوجهات في الشبكة. من هذا المنطلق يمكن صياغة مشكلة التوجيه بطريقة رياضية مثل مشكلة تحقيق حل أمثل ذات محدودات (constrained optimization problem) والمعروفة بمشكلة تدفق الشبكة (network flow problem [Bertsekas 1991]. نذكر هنا نوعاً آخر من خوارزميات التوجيه مشتقّ من عالم الإرسال الهاتفي يعرف بخوارزميات التوجيه بتحويل الدوائر، ولهذا النوع أهمية في

شبكات البيانات بتحويل الرزم في الحالات التي نحتاج فيها لحجز موارد الوصلة (مثل الذاكرة المؤقتة والحيز الترددي للوصلة) لكل توصيلة تمر خلال الوصلة. بينما قد تبدو صياغة مشكلة التوجيه بهذا الأسلوب مختلفة تماماً عن صياغة توجيه أدنى كلفة التي رأيناها في هذا الفصل، إلا أن هناك عدد من التشابهات على الأقل من حيث خوارزمية إيجاد المسار (خوارزمية التوجيه). اطّلِع على [Ash 1998; Ross 1995; Girard 1990] لمناقشة أكثر تفصيلاً لأبحاث في هذا الموضوع.

3-5-4 التوجيه الهرمي (Hierarchical Routing)

في دراستنا لخوارزميات LS وDV نظرنا للشبكة ببساطة كمجموعة من الموجّهات المرتبطة ببعضها البعض. لم نُفرِّق بين موجّه وآخر حيث إنها جميعاً تُنفِّذ نفس خوارزمية التوجيه لحساب مسارات التوجيه خلال الشبكة بكاملها. عملياً هذا النموذج ونظرته للموجّهات كمجموعة متجانسة يُنفّذ كلٌّ منها نفس خوارزمية التوجيه يُعد تبسيطاً للأمور إلى حد كبير، وذلك لسببين مهمين:

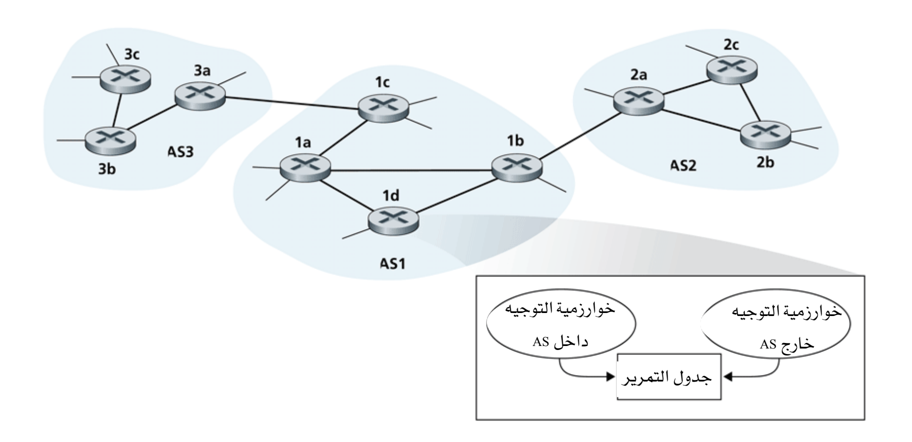
- حجم الشبكة: مع زيادة عدد الموجّهات يصبح العبء الإضافي في حساب وتخزين وتبادل معلومات التوجيه (على سبيل المثال رسائل التحديث في LS وتغييرات المسار الأدنى كلفة) عائقاً. تشمل الإنترنت العامّة اليوم مئات الملايين من المضيفات، وواضح أن تخزين معلومات التوجيه في كل تلك المضيفات يتطلب كميات هائلة من الذاكرة. وكذلك فإن العبء الإضافي لإذاعة التحديثات في LS بين كل الموجّهات في الإنترنت العامّة لن يترك أي حيز ترددي لإرسال رزم البيانات! وبالتأكيد فإن خوارزمية متجه المسافة التي تتكرّر بين مثل هذا العدد الكبير من الموجّهات لن تتقارب أبداً. واضح أنه يجب عمل شيء لتخفيض تعقيد حساب المسار في الشبكات الكبيرة كالإنترنت العامّة.
- ذاتية الحكم الإداري: يميل الباحثون لإهمال بعض القضايا كرغبة شركة ما في تشغيل موجّهاتها كما يحلو لها (على سبيل المثال تشغيل خوارزمية التوجيه التي تختارها) أو في إخفاء السمات التنظيمية للشبكة الداخلية عن خارج الشبكة، ولكن تلك في واقع الأمر اعتبارات مهمة. فمن الناحية

المثالية يجب أن تكون المنظمة قادرة على تشغيل وإدارة شبكتها كما تحب، وفي الوقت ذاته تبقى قادرة على توصيل شبكتها بالشبكات الأخرى الخارجية.

يمكن أن تُحل كلتا هاتين المشكلتين بتنظيم الموجّهات في أنظمة مستقلة ذاتياً ((Autonomous Systems (ASs))، يتكون كلٌّ منها من مجموعة من الموجّهات التي عادةً ما تكون تحت نفس الرقابة الإدارية (مثلاً تُشغَّل من قِبَل نفس موفِّر خدمة الإنترنت أو تنتمي لشبكة شركة بعينها). تُنفِّذ جميع الموجّهات داخل النظام المستقل ذاتياً نفس خوارزمية التوجيه (كخوارزمية LS أو DV) وتتوافر لديها المعلومات عن بعضها البعض، بالضبط كما في حالة النموذج المثالي الذي تناولناه في الجزء السابق. يطلق على خوارزمية التوجيه التي تعمل ضمن نظام مستقل ذاتياً بروتوكول توجيه داخل نظام مستقل ذاتياً (intra-AS routing). بالطبع سيكون من الضروري توصيل تلك الأنظمة المستقلة ذاتياً إلى بعضها البعض، وبالتالي سيضاف موجّه أو أكثر تناط به المسؤولية الإضافية لتوجيه الرزم إلى الوجهات خارج النظام المستقل ذاتياً (inter-AS routing)؛ تسمى تلك الموجّهات موجّهات البوّابة (gateway .(routers

يبين الشكل 4-32 مثالاً بسيطاً لثلاثة أنظمة مستقلة ذاتياً: AS1، AS2، وAS3. تمثّل الخطوط الثقيلة في هذا الشكل الوصلات المباشرة بين أزواج الموجّهات. أما الخطوط الأخف بين الموجّهات فتمثّل الشبكات الفرعية التي تتصل مباشرة بتلك الموجّهات. يتكون AS1 من أربعة موجّهات (la، 1b، lc) تدير عملية التوجيه داخل AS1. ولذا فكلٌّ من هذه الموجّهات الأربعة تعرف كيف ترسل الرزم على طول المسار الأمثل (optimal route) إلى أي وجهة ضمن AS1. وبنفس الطريقة لكلٍّ من الأنظمة المستقلة ذاتياً AS2 وAS3 ثلاثة موجّهات. لاحظ أن بروتوكولات التوجيه داخل الأنظمة AS1 وAS2 وAS3 لا يشترط أن تكون هي نفسها. لاحظ أيضاً أن الموجّهات 1b و1c و2a و3a تعمل كموجّهات بوّابة.

نأمل أن يكون واضحاً لديك الآن كيف تقوم الموجّهات في AS بتحديد مسارات التوجيه بين أزواج المصدر والوجهة الموجودة داخل AS. لكن ما زالت هناك قطعة كبيرة مفقودة من لغز التوجيه من طرف إلى طرف. كيف لموجّه يعمل داخل AS أن يعرف كيف يوجّه رزماً إلى وجهة خارج AS؟ من السّهل الإجابة على هذا السؤال إذا كان نظام AS لديه موجّه بوّابة واحد يوصله إلى AS آخر فقط. في هذه الحالة تحدد خوارزمية التوجيه داخل AS مسار أدنى كلفة من كل موجّه داخلي إلى موجّه البوّابة، والذي يعرف بدوره كيف يوجه الرزمة. بمجرد استلام موجّه البوّابة الرزمة فإنه يرسلها على الوصلة الواحدة التي تقود إلى خارج AS. يتحمل نظام AS على الجانب الآخر من الوصلة مسؤولية توجيه الرزمة إلى الوجهة النهائية. وكمثال افترض أن الموجّه 2b في الشكل 4-32 يستلم رزمة وجهتها خارج AS2. سيرسل الموجّه 2b بعد ذلك الرزمة لأيٍّ من الموجّهات 2a أو 2c حسبما هو مبين بجدول التمرير للموجّه 2b، والذي أُعد من قِبَل بروتوكول التوجيه داخل AS2. ستصل الرزمة في النهاية إلى موجّه البوّابة 2a، والذي سيرسلها إلى 1b. بمجرد مغادرة الرزمة 2a تكون مهمة AS2 قد انتهت مع تلك الرزمة.



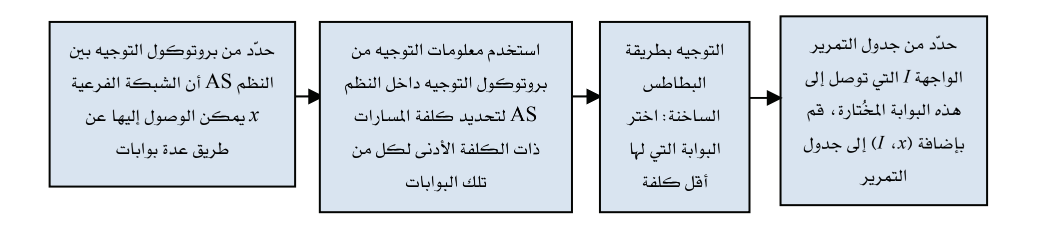
الشكل 4-32 مثال لعدة أنظمة مستقلة ذاتياً متصلة ببعضها.

ولذا فالمشكلة سهلة عندما يكون لـ AS المصدر وصلة واحدة فقط تقود خارجه. لكن ماذا لو أن AS المصدر كانت له وصلتان أو أكثر (خلال موجّهي بوّابة أو أكثر) تقود خارجه؟ عندئذ تصبح مشكلة معرفة أين تُرسل الرزمة أكثر تحدّياً. على سبيل المثال خذ في الاعتبار موجّهاً في AS1 كان قد تلقى رزمة وجهتها خارج AS1. واضح أنه يجب على الموجّه أن يرسل الرزمة إلى إحدى موجهات البوّابة 1b أو 1c. لكن لأيٍّ منهما؟ لحلّ هذه المشكلة يحتاج AS1 إلى (1) معرفة أي الوجهات يمكن الوصول إليها عن طريق AS2 وأيٍّ منها يمكن الوصول إليه عن طريق AS3، و(2) نشر معلومات الوصول هذه إلى كل الموجهات داخل AS1 حتى يمكن لكل موجّه أن يعد جدول تمرير لمعالجة الوجهات الخارجية. يتم أداء هاتين المهمّتين (الحصول على معلومات الوصول من نظم AS المجاورة ونشر تلك المعلومات لكل الموجّهات الداخلية) بواسطة بروتوكول التوجيه بين الأنظمة المستقلة ذاتياً. ولأن بروتوكول التوجيه بين ASs يتضمّن الاتصال بين نظامي AS، فمن الضروري أن يستخدم النظامان نفس بروتوكول التوجيه البيني. في الحقيقة تستخدم كل ASs في الإنترنت نفس بروتوكول التوجيه بين الأنظمة المستقلة ذاتياً، والمعروف ببروتوكول BGP4، والذي سنناقشه في الجزء التالي. كما هو موضح في الشكل 4-32 يستلم كل موجّه معلومات من بروتوكول التوجيه داخل AS ومن بروتوكول التوجيه خارج AS، ويستخدم المعلومات من كليهما لإعداد جدول التمرير.

وكمثال افترض شبكة فرعية x (مُعرّفة بعنوان CIDR)، وافترض بأن AS1 علِم من بروتوكول التوجيه بين ASs أنه يمكن الوصول لتلك الشبكة الفرعية x عن طريق AS3 ولكن لا يمكن الوصول لها عن طريق AS2. عندئذ يذيع AS1 هذه المعلومات إلى كل موجّهاته. عندما يعلم الموجّه 1d بأن الشبكة الفرعية x يمكن الوصول لها من AS3 ومن ثمَّ من موجّه البوّابة 1c، عندئذ يقرّر من المعلومات المتوفرة من قِبَل بروتوكول التوجيه بين ASs واجهة الموجّه التي على مسار أدنى كلفة من الموجّه 1d إلى موجّه البوّابة 1c. افترض أن هذه الواجهة هي I. يمكن أن يضيف الموجّه 1d المدخل (x, I) إلى جدول التمرير لديه. (هذا المثال وأمثلة أخرى قدمت في هذا الجزء تبين الفكرة العامّة لكنها تبسيط لما يحدث فعلاً في الإنترنت. في الجزء

التالي سنزوّدك بوصف أكثر تفصيلاً ولو أنه أكثر تعقيداً عندما نناقش بروتوكول BGP).

وتعقيباً على المثال السابق افترض الآن أن AS2 وAS3 متصلان بـ ASs أخرى غير موضحة في الشكل. افترض أيضاً أن AS1 علِم من بروتوكول التوجيه بين ASs أن الشبكة الفرعية x يمكن الوصول لها من AS2 (عن طريق موجّه البوّابة 1b) ومن AS3 (عن طريق موجّه البوّابة 1c). عندئذ سيذيع AS1 هذه المعلومات إلى كل الموجّهات بداخله بما في ذلك d1. ولكي ينشئ الموجّه 1d جدول التمرير لديه عليه أن يحدد مَنْ من الموجهين 1b أو 1c يجب أن يوجّه الرزم المتجهة إلى x. أحد الطرق التي غالباً ما تُستخدَم عملياً تتبع أسلوب توجيه البطاطس الساخنة (hot-potato routing). في هذه الطريقة يتخلص AS من الرزمة (البطاطس الساخنة) بأسرع ما يمكن (بدقة أكثر بأرخص ما يمكن). يتم ذلك بجعل الموجّه يرسل الرزمة إلى موجّه البوّابة الذي يعتبر المسار إليه له أدنى كلفة ويتوفر لديه مسار إلى الوجهة المطلوبة. في سياق المثال الحالي يستخدم الموجّه 1d توجيه البطاطس الساخنة والذي يستعمل معلومات من بروتوكول التوجيه داخل AS لتحديد كلفة المسارات إلى 1b و1c، وحينئذ سيختار المسار الذي له أدنى كلفة. بمجرد اختيار هذا المسار يضيف الموجّه 1d مدخلاً جديداً للشبكة الفرعية x في جدول التمرير لديه. يلخّص الشكل 4-33 ما يفعله الموجّه 1d لإضافة المدخل الجديد لـ x إلى جدول التمرير.



الشكل 4-33 خطوات إضافة وجهة خارج AS في جدول تمرير الموجّه.

عندما يعلم AS عن وجهة من نظام AS مجاور يمكن أن يعلن AS معلومات التوجيه تلك لبعض ASs أخرى مجاورة له. على سبيل المثال افترض أن AS1 يعلم من AS2 أن الشبكة الفرعية x يمكن الوصول لها عن طريق AS2. يمكن لـ AS1 أن يخبر AS3 أنه يمكن الوصول لـ x عن طريق AS1. بهذه الطريقة إذا كانت AS3 تحتاج لتوجيه رزمة إلى x فإن AS3 سيرسل الرزمة إلى AS1 والذي بدوره يرسلها إلى AS2. كما سنرى في مناقشتنا لبروتوكول BGP أن لـ AS مرونة في تحديد أيٍ من الوجّهات سيعلن عنها لنظم ASs المجاورة. هذا قرار تمليه سياسة التشغيل المتبعة، ويعتمد عادةً على القضايا الاقتصادية أكثر من اعتماده على القضايا التقنية.

تذكّر من الجزء 1-5 أن الإنترنت تتكون من مزوّدي خدمة الإنترنت الموزعين في تركيب هرمي. إذاً ما العلاقة بين موفري خدمة الإنترنت وASs؟ قد تعتقد بأن الموجّهات التابعة لموفر خدمة إنترنت والوصلات التي تربط بينها تُعتبر بمثابة نظام AS واحد. بالرغم من أن الأمر يكون كذلك في أغلب الأحيان، إلا أن العديد من موفري خدمة الإنترنت يقسمون شبكتهم إلى عدة ASs. على سبيل المثال بعض موفري خدمة الإنترنت من الطبقة الأولى (tier-1) يصممون شبكتهم بكاملها كنظام AS واحد، بينما يقسم البعض الآخر شبكته إلى عشرات من أنظمة ASs المتصلة ببعضها.

الخلاصة أنه يتم حل مشاكل حجم الشبكة والسلطة الإدارية بتعريف الأنظمة المستقلة ذاتياً. تستخدم كل الموجّهات الموجودة داخل AS نفس بروتوكول التوجيه داخله، في حين تستخدم كل أنظمة ASs المختلفة نفس بروتوكول التوجيه فيما بينها. تُحل مشكلة حجم الشبكة بهذه الطريقة لأن بروتوكول التوجيه داخل AS يحتاج فقط لمعرفة الموجّهات داخل AS. وتحل مشكلة السلطة الإدارية لأن أي منظمة يمكن أن تستخدم أي بروتوكول توجيه تختاره داخل نظام AS الخاص بها؛ لكن كل زوج من ASs المتصلة ببعضهما يحتاج لتشغيل نفس بروتوكول التوجيه بين ASs لتبادل معلومات الوصول (reachability).

في الجزء التالي سنتناول ثلاثة من البروتوكولات المستخدمة في الإنترنت اليوم: اثنين للتوجيه داخل AS (RIP وOSPF) وبروتوكولاً للتوجيه بين ASs (BGP). بدراسات الحالة تلك ستكتمل دراستنا للتوجيه الهرمي بشكلٍ جيد.

4-6 التوجيه في الإنترنت

بعد أن درسنا عناوين الإنترنت وبروتوكول IP، نحوّل انتباهنا الآن إلى بروتوكولات التوجيه في الإنترنت (وهي المسؤولة عن تحديد المسارات التي تأخذها وحدات البيانات من المصدر إلى الوجهة). سنرى أن بروتوكولات التوجيه في الإنترنت تجسّد العديد من المبادئ التي تعلّمناها في وقت سابق في هذا الفصل. إن طرق حالة الوصلة ومتجه المسافة التي درسناها في الأجزاء 4-5-1 و4-5-2 وفكرة الأنظمة المستقلة ذاتياً في الجزء 4-5-3 تمثل جميعاً محاور مركزية للتوجيه في الإنترنت اليوم.

تذكّر من الجزء 4-5-3 أن النظام المستقل ذاتياً (AS) هو مجموعة من الموجّهات تحت نفس السلطة الإدارية والتقنية، وتستخدم جميعها نفس بروتوكول التوجيه فيما بينها. وبدوره يحتوى كل AS عادة على عدة شبكات فرعية (راجع الاستعمال الدقيق لتعبير شبكة فرعية مع العنونة في الجزء 4-4-2).

4-6-1 التوجيه داخل نظام AS في الإنترنت: بروتوكول RIP

يُستخدم بروتوكول التوجيه داخل AS لتحديد المسارات داخل نظام مستقل ذاتيا (AS). تعرف بروتوكولات التوجيه داخل AS أيضاً ببروتوكولات البوّابة الداخلية (interior gateway protocols). من الناحية التاريخية اُستخدِم اثنان من بروتوكولات التوجيه على نطاق واسع للتوجيه ضمن نظام مستقل ذاتياً في الإنترنت: بروتوكول معلومات التوجيه RIP (Routing Information Protocol) وبروتوكول المسار الأقصر أولاً المفتوح OSPF (Open Shortest Path First). وهناك بروتوكول توجيه آخر وثيق الصلة بـ OSPF يعرف بـ IS-IS [RFC 1142; Perlman 1999]. سنناقش أولاً RIP وبعد ذلك OSPF.

يُعتبر RIP أحد البروتوكولات الأولى للتوجيه داخل AS في الإنترنت وما زال واسع الانتشار اليوم. ترجع أصوله واسمه إلى البنية المعمارية لأنظمة شبكة Xerox (XNS)، ويعزى استخدامه على نطاق واسع الانتشار بدرجة كبيرة إلى إدراجه ضمن نسخة BSD لنظام التشغيل يونيكس UNIX بدعم لبروتوكولات TCP/IP والتي ظهرت عام 1982. تم تعريف النسخة 1 من بروتوكول RIP في [RFC 1058]، والنسخة 2 بتوافق خلفي في [RFC 2453].

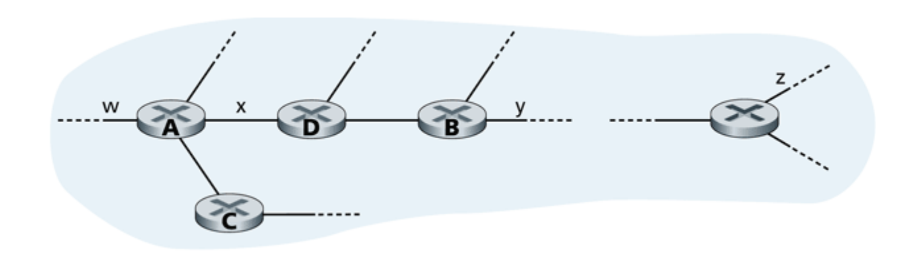
يُعد RIP بروتوكول متجه المسافة ويعمل بأسلوب قريب جداً من الحالة المثالية لبروتوكول DV الذي فحصناه في الجزء 4-5-2. تُستخدم نسخة RIP المعرَّفة في RFC 1058 عدد القفزات كمعيار للكلفة؛ أي أن كل وصلة لها كلفة تساوي 1. في خوارزمية DV في الجزء 4-5-2 وللتبسيط كنا قد عرّفنا الكلفة بين أزواج الموجّهات. في RIP (وأيضاً في OSPF) تعرّف الكلفة في الحقيقة من موجّه المصدر إلى شبكة الوجهة الفرعية. يستخدم RIP المصطلح قفزة (hop) للإشارة إلى عدد الشبكات الفرعية التي يتم عبورها على طول المسار الأقصر من موجّه المصدر إلى شبكة الوجهة الفرعية، بما في ذلك شبكة الوجهة الفرعية. يوضح الشكل 4-34 نظام AS بست شبكات فرعية طرفية. يبين الجدول الموجود بالشكل عدد القفزات من المصدر A إلى كلٍّ من الشبكات الفرعية الطرفية.

الواجهة	عدد القفزات
u	1
v	2
w	2
x	3
y	3
z	2

الشكل 4-34 عدد القفزات من موجّه المصدر A إلى الشبكات الفرعية المختلفة.

تم تحديد الكلفة القصوى لمسارٍ ما بالعدد 15، وبالتالي يمكن استخدام RIP فقط للأنظمة المستقلة ذاتياً التي لا يزيد طول قطرها عن 15 قفزة. تذكّر أن الموجّهات المتجاورة في بروتوكولات DV تتبادل متجهات المسافة مع بعضها البعض. يمثل متجه المسافة لأيٍّ من تلك الموجّهات التقدير الحالي لأطوال المسارات الأقصر بين ذلك الموجّه والشبكات الفرعية في AS. يتم تبادل رسائل تحديث التوجيه في RIP بين الجيران كل 30 ثانية تقريباً عن طريق رسالة ردّ RIP. تحتوي رسالة الردّ التي يرسلها موجّه أو مضيف قائمةً لا يزيد طولها عن 25 شبكة وجهة فرعية ضمن AS، بالإضافة إلى المسافة من المُرسِل إلى كلٍّ من تلك الشبكات الفرعية. تعرف رسائل الرد أيضاً بإعلانات RIP (RIP advertisements).

لنلقِ نظرةً على مثال بسيط لكيفية عمل إعلانات RIP. خذ في الاعتبار جزءاً من نظام AS كالموضح في الشكل 4-35. في هذا الشكل تدل الخطوط التي توصل بين الموجّهات على الشبكات الفرعية. فقط تم تسمية بعض الموجّهات المختارة (A، B، C، D) والشبكات الفرعية (z ،w , x, y). تشير الخطوط المنقطة بأن AS ممتد؛ وهكذا يتكون هذا النظام المستقل ذاتياً من المزيد من الموجّهات والوصلات بالإضافة إلى تلك الموضحة بالشكل.



الشكل 4-35 جزء من نظام مستقل ذاتياً.

عدد القفزات	الموجّه التالي	الشبكة الفرعية للوجهة
2	A	w
2	B	y
7	B	z
1	--	x
...	...	...

الشكل 4-36 جدول التوجيه في الموجّه D قبل استلام الإعلان من الموجّه A.

يحتفظ كل موجّه بجدول RIP يُعرَف باسم جدول التوجيه. يشتمل جدول التوجيه لدى الموجّه على كلٍّ من متجه المسافة للموجّه وجدول التمرير للموجّه. يوضح الشكل 4-36 جدول التوجيه للموجّه D. لاحظ أن جدول التوجيه يتكون من ثلاثة أعمدة. يشير العمود الأول إلى شبكة الوجهة الفرعية، ويشير العمود الثاني إلى عنوان الموجّه التالي على طول المسار الأقصر إلى شبكة الوجهة الفرعية، ويشير العمود الثالث إلى عدد القفزات (أي عدد الشبكات الفرعية التي يجب أن تُعبر بما في ذلك شبكة الوجهة الفرعية) للوصول إلى شبكة الوجهة الفرعية على طول المسار الأقصر. في هذا المثال يبين الجدول أنه لكي ترسل رزمة بيانات من الموجّه D إلى شبكة الوجهة الفرعية *w*، يجب أن ترسل رزمة البيانات أولاً إلى الموجّه المجاور A؛ وأن شبكة الوجهة الفرعية *w* تبعد مسافة قفزتين على طول المسار الأقصر. بنفس الطريقة يبين الجدول أن الشبكة الفرعية *z* تبعد مسافة 7 قفزات عن طريق الموجّه B. من حيث المبدأ سيحتوي جدول التوجيه على صف واحد لكل شبكة فرعية في نظام AS ، رغم أن النسخة 2 من RIP تسمح بتجميع (تكتيل) مدخلات الشبكة الفرعية باستعمال تقنيات تشبه تقنيات تجميع المسارات التي تناولناها في الجزء 4-4. الجدول في الشكل 4-36 والجداول التي ستأتي لاحقاً هي جداول جزئية فقط.

افترض الآن أنه بعد 30 ثانية يستلم الموجّه D من الموجّه A الإعلان المبين في الشكل 4-37. لاحظ أن هذا الإعلان ما هو إلا معلومات جدول التوجيه من الموجّه A! تشير تلك المعلومات بشكلٍ خاص إلى أن الشبكة الفرعية z تبعد فقط أربع

قفزات عن الموجّه A. فور استلام الموجّه D لهذا الإعلان يدمج الإعلان (الشكل 4-37) بجدول التوجيه القديم (الشكل 4-36). وبشكل خاص يعرف الموجّه D أن هناك الآن مسار من خلال الموجّه A إلى الشبكة الفرعية z أقصر من المسار من خلال الموجّه B. وبالتالي يُحدّث الموجّه D جدول التوجيه لديه ليشمل المسار الأقصر كما هو موضح في الشكل 4-38. قد تتساءل: وكيف يصبح المسار الأقصر إلى الشبكة الفرعية z أقصر؟ من المحتمل أن خوارزمية متجه المسافة اللامركزية ما زالت في عملية التقارب (انظر الجزء 4-5-2)، أو ربما قد أضيفت وصلات جديدة أو موجّهات جديدة أو كلاهما إلى AS، ومن ثمَّ تغيّر المسار الأقصر في AS.

الشبكة الفرعية للوجهة	الموجّه التالي	عدد القفزات
z	C	4
w	--	1
x	--	1
...	...	...

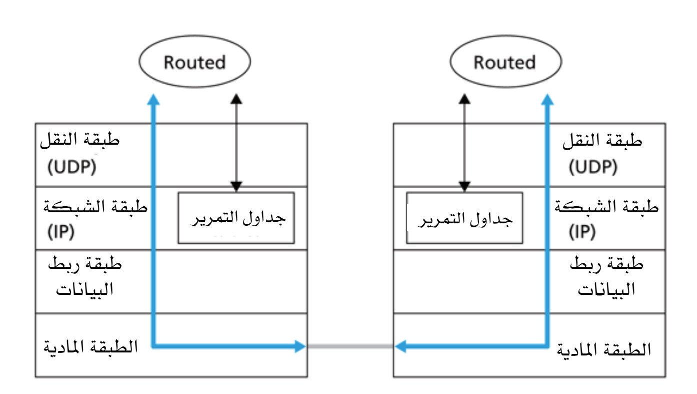
الشكل 4-37 الإعلان من الموجّه A.

الشبكة الفرعية للوجهة	الموجّه التالي	عدد القفزات
w	A	2
y	B	2
z	A	5
...	...	...

الشكل 4-38 جدول التوجيه في الموجه D بعد استلام الإعلان من الموجّه A.

دعنا نستعرض بضع سمات تطبيقية لبروتوكول RIP. تذكر أن موجّهات RIP تتبادل الإعلانات كل 30 ثانية تقريباً. إذا لم يسمع الموجّه من أحد جيرانه على الأقل مرة كل 180 ثانية فسيعتبر أن ذلك الجار لم يعد يستطيع الوصول إليه؛ أي أن ذلك الجار قد مات أو أن الوصلة بينهما قد انقطعت. عندما يحدث ذلك يعدّل بروتوكول RIP جدول التوجيه المحلي وبعد ذلك يذيع تلك المعلومات بإرسال الإعلانات إلى الموجّهات المجاورة (تلك التي ما زال في الإمكان الوصول إليها). يمكن أن يطلب موجّه أيضاً معلومات حول كلفة أحد الجيران للوصول لوجهة معينة باستخدام رسالة طلب RIP. ترسل الموجهات رسائل طلب RIP ورسائل ردّ RIP إلى بعضها البعض باستخدام منفذ UDP رقم 520. تُنقل قطعة UDP بين الموجّهات في رزمة بيانات IP قياسية. في الحقيقة إن RIP يستخدم بروتوكول طبقة النقل UDP فوق بروتوكول طبقة الشبكة IP لتنفيذ وظيفة طبقة الشبكة (خوارزمية التوجيه) قد يبدو ملتوياً بعض الشيء (وهو فعلاً كذلك!). ولو نظرنا بعمق أكثر إلى كيفية تحقيق RIP لاتضح لنا هذا اللبس.

يبين الشكل 4-39 كيف يُطبّق RIP عادةً في نظام التشغيل يونيكس (UNIX) (على سبيل المثال في محطة عمل يونيكس تعمل كموجّه). تُنفذ عملية تسمى routed (وتنطق روت - دي) بروتوكول RIP، أي تحتفظ بمعلومات التوجيه وتتبادل الرسائل مع عمليات التوجيه التي تعمل في الموجّهات المجاورة. نظراً لأن بروتوكول RIP يُنفّذ كعملية في طبقة التطبيقات (ولو أنها عملية خاصّة جداً حيث لديها القدرة على معالجة جداول التوجيه بداخل لب اليونيكس UNIX kernel)، يكون بوسع البروتوكول إرسال وتلقّي رسائل على مقبس قياسي وكذلك استخدام بروتوكول نقل قياسي. كما وضحنا يُنفّذ RIP كبروتوكول طبقة تطبيقات (انظر الفصل الثاني) يعمل فوق UDP.



الشكل 4-39 تطبيق RIP كعملية routed daemon في نظام التشغيل يونيكس.

4-6-2 التوجيه داخل AS في الإنترنت: بروتوكول OSPF

كما هو الحال مع بروتوكول RIP، يُستخدم بروتوكول OSPF على نطاق واسع للتوجيه في الإنترنت داخل أنظمة AS. يستعمل OSPF وبروتوكول آخر وثيق الصلة به (IS-IS) عادةً في شبكات موفري خدمة الإنترنت من الطبقة العليا (upper-tier)، بينما يستعمل RIP في شبكات موفري خدمة الإنترنت من الطبقة الدنيا (lower-tier) وشبكات الشركات. تدل كلمة open في اسم البروتوكول على أنه متوفر للجميع (ملكية عامة) (في مقابل الملكية الخاصة كما في حالة بروتوكول سيسكو EIGRP). تُعتبر النسخة 2 هي أحدث نسخة من OSPF وهي معرّفة في RFC 2328.

يعتبر OSPF أيضاً بمثابة الوريث لبروتوكول RIP ولذا فإنه يمتاز بعدة ميزات متقدّمة. ومع ذلك يعتبر OSPF في صميمه بروتوكول حالة الوصلة LS الذي يستعمل فيضان معلومات حالة الوصلة وخوارزمية Dijkstra لحساب المسارات الأدنى كلفة. باستخدام بروتوكول OSPF يبني الموجّه خريطة طوبوغرافية كاملة (أي رسماً بيانياً) للنظام المستقل ذاتياً AS بكامله. بعد ذلك يشغّل الموجّه خوارزمية Dijkstra محلياً لحساب شجرة أقصر المسارات إلى كل الشبكات الفرعية والتي يمثل الموجّه نفسه عقدة الجذر لها. يتم إعداد كلفة كل وصلة منفردة من قبل المشرف على الشبكة (انظر المبادئ والواقع العملي: إعداد أوزان الوصلات في بروتوكول

OSPF). قد يختار مشرف الشبكة قيمة الكلفة 1 لكل وصلة، ومن ثم ينجز توجيه أدنى القفزات، أو قد يختار تحديد أوزان الوصلة بطريقة تتناسب عكسياً مع سعة الوصلة لكي يقلل من مرور البيانات خلال الوصلات ذات سعة الإرسال المنخفضة. لا يشترط OSPF سياسة معينة لتحديد أوزان الوصلات (فتلك مهمة المشرف على الشبكة)، لكن بدلاً من ذلك يوفر OSPF الآليات (على شكل بروتوكول) لحساب المسارات الأدنى كلفة لمجموعة أوزان معطاة.

في بروتوكول OSPF يذيع الموجّه معلومات التوجيه لدعوة الموجّهات الأخرى في النظام المستقل ذاتياً (وليست الموجّهات المجاورة فقط). يذيع الموجّه معلومات حالة الوصلة حينما يكون هناك تغيّر في حالة وصلة (مثلاً تغيّر في الكلفة أو تغيّر في الحالة من كونها متعطّلة إلى شغّالة والعكس). أيضاً يذيع الموجّه حالة الوصلة بشكلٍ دوري (على الأقل مرة كل 30 دقيقة) حتى إذا لم يحدث تغيّر في حالة الوصلة. ينص RFC 2328 على أن "هذا التجديد الدوري بإعلانات حالة الوصلة يضيف متانة (موثوقية) لخوارزمية حالة الوصلة". توضع إعلانات OSPF في رسائل OSPF والتي تُنقل مباشرة من قِبَل IP حيث يستخدم الرقم 89 (أي OSPF) لتمثيل بروتوكول الطبقة الأعلى. وبالتالي يجب أن يُنفّذ بروتوكول OSPF نفسه الوظائف المختلفة كنقل الرسالة الموثوق فيه وإذاعة حالة الوصلة. يتأكد OSPF أيضاً من أن حالة الوصلة شغّالة (عن طريق رسالة HELLO "مرحباً" التي تُرسَل إلى أحد الجيران المرتبطين) وتسمح لموجّه OSPF بالحصول على قاعدة بيانات أحد الموجّهات المجاورة والتي تتضمن حالة الوصلات في كافة أنحاء الشبكة.

يتضمّن بروتوكول OSPF عدة خصائص متطورة من بينها:

- الأمن: يمكن توثيق تبادل المعلومات بين موجّهات OSPF (مثلاً لتحديث حالة الوصلة). ويسمح التوثيق للموجّهات الموثوق فيها فقط بالمشاركة ضمن بروتوكول OSPF داخل أنظمة AS وهكذا يُمنَع المتطفلون المؤذيون (أو طلاب الشبكات الذين يجربون ما تعلموه حديثاً للتسلية) من حقن معلومات غير صحيحة في جداول التوجيه. وبشكلٍ اعتيادي لا يستخدم OSPF التوثيق للتحقّق من رزم OSPF التي تنتقل بين الموجّهات ولذا يمكن تزييفها. يمكن

استخدام نوعين من التوثيق: بسيط وMD5 (انظر الفصل الثامن لمناقشة MD5 والتوثيق بصفة عامة). يعني التوثيق البسيط استخدام نفس كلمة السر على كل الموجّهات. فعندما يرسل موجّه OSPF رزمة يُضَمّن كلمة السر في النص الأصلي (plaintext). من الواضح أن التوثيق البسيط غير آمن تماماً. يعتمد MD5 على المفاتيح السرية المشتركة التي يتم إعدادها في كل الموجّهات. يحسب الموجّه ملخص MD5 (MD5 hash) لكل رزمة OSPF يرسلها بناءً على محتوى الرزمة والمفتاح السري الذي يذيلها (انظر مناقشة أكواد توثيق الرسائل في الفصل السابع). بعد ذلك يضع الموجّه قيمة الملخص الناتج ضمن رزمة OSPF. يستخدم موجّه الاستقبال المفتاح السري المعد من قبل ويحسب ملخص MD5 من الرزمة ويقارنه بقيمة ملخص MD5 الذي تحمله الرزمة، وهكذا يتحقّق من مصداقية الرزمة. تستخدم أيضاً الأرقام التسلسلية مع MD5 لتوفير الحماية ضدّ هجوم إعادة التشغيل (replay attack).

- استخدام مسارات متعدّدة بنفس الكلفة: عندما تكون هناك مسارات متعدّدة لها نفس الكلفة إلى وجهة ما يسمح OSPF باستعمال عدة مسارات (بمعنى أنه ليس من الضروري أن يختار مساراً وحيداً لنقل كل حركة مرور البيانات عندما توجد عدة مسارات ذات كلفة متساوية).
- الدعم المتكامل لتوجيه الإرسال الفردي والمتعدّد: يوفر بروتوكول MOSPF للإرسال المتعدد (multicast OSPF) [RFC 1584] امتدادات بسيطة لبروتوكول OSPF لتوجيه الإرسال المتعدد (سنغطّي هذا الموضوع بعمق أكثر في الجزء 4-7-2). يستخدم MOSPF قاعدة بيانات الوصلات الخاصة بـ OSPF ويضيف نوعاً جديداً من إعلان حالة الوصلة لآلية إذاعة حالة الوصلة في OSPF.
- دعم التوجيه الهرمي داخل نفس نطاق التوجيه: لعل أهم جوانب التقدم التي حققها بروتوكول OSPF هي قدرته على تنظيم نظامٍ مستقلٍ ذاتياً بشكلٍ هرمي. رأينا في الجزء 4-5-3 العديد من المزايا لتراكيب التوجيه الهرمي. سنغطّي تطبيق التوجيه الهرمي لـ OSPF في بقيّة هذا الجزء.

باستخدام OSPF يمكن تقسيم النظام المستقل ذاتياً إلى مناطق (areas). تستخدم كل منطقة خوارزمية توجيه حالة الوصلة الخاصة بها، ويذيع كل موجّه في منطقة حالة وصلاته إلى باقي الموجّهات في تلك المنطقة. بهذه الطريقة تبقى التفاصيل الداخلية لمنطقةٍ ما مخفية عن كل الموجّهات خارج تلك المنطقة. يتضمّن التوجيه داخل المنطقة (intra-area routing) فقط الموجّهات داخل المنطقة نفسها.

وضمن كل منطقة يضطلع واحدٌ أو أكثر من الموجّهات الموجودة على حدود المنطقة بمهمة توجيه الرزم خارج المنطقة. يتم تهيئة منطقة واحدة فقط من المناطق لكي تكون منطقة العمود الفقري. ويكون الدور الأساسي لمنطقة العمود الفقري هو توجيه مرور البيانات بين المناطق الأخرى في نظام AS. يحتوي العمود الفقري دائماً على موجّهات الحدود للمناطق في AS بالإضافة إلى احتمال وجود موجّهات أخرى غير حدودية.

يتطلّب التوجيه بين المناطق داخل AS توجيه الرزمة أولاً إلى موجّهٍ موجود على حدود منطقة المصدر (توجيه داخل منطقة)، ثمّ توجيهها عبر العمود الفقري إلى موجّه حدود المنطقة التي فيها الوجهة، ومن ثم توجّه داخل تلك المنطقة للوجهة النهائية.

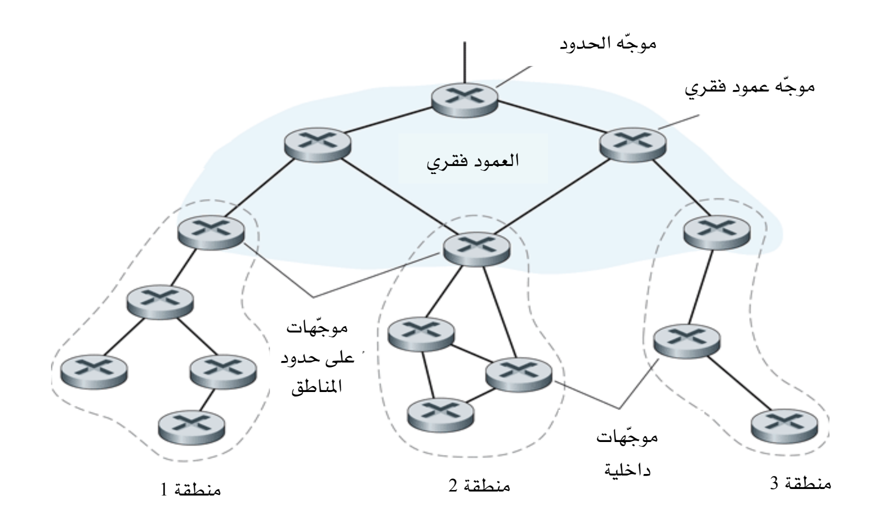
يوضح الشكل 4-40 مخططاً لتركيب هرمي لشبكة OSPF. يمكن أن نميّز أربعة أنواع من موجّهات OSPF في الشكل:

- موجّهات داخلية (internal routers): تقع تلك الموجّهات في مناطق غير العمود الفقري وتؤدّي مهمة التوجيه داخل AS فقط.
- موجّهات حدود منطقة (area border routers): تنتمي تلك الموجّهات إلى أحد المناطق بالإضافة إلى العمود الفقري.
- موجّهات عمود فقري (موجّهات غير حدودية) (backbone routers): تؤدّي تلك الموجّهات مهمة التوجيه داخل العمود الفقري، لكنها ليست في حد ذاتها موجّهات حدود منطقة. داخل منطقة غير العمود الفقري، تعرف الموجّهات الداخلية بوجود مسارات إلى المناطق الأخرى من المعلومات المذاعة داخل المنطقة من موجّهات العمود الفقري بها (بشكلٍ جوهري من إعلانات حالة

الوصلة، لكنها تعلن كلفة المسار إلى المنطقة الأخرى بدلاً من كلفة الوصلة).

- موجّهات حدود AS (boundary routers): تتبادل الموجّهات على حدود نظام AS معلومات التوجيه مع الموجّهات في الأنظمة الأخرى المستقلة ذاتياً. قد تستخدم تلك الموجّهات مثلاً بروتوكول BGP لعملية التوجيه بين الأنظمة المستقلة ذاتياً. من خلال مثل تلك الموجّهات تتعلّم الموجّهات الأخرى المسارات إلى الشبكات الخارجية.

يعتبر OSPF بروتوكولاً معقّداً نسبياً، وتغطيتنا له هنا كانت بالضرورة مقتضبة؛ يمكنك الاطلاع على تفاصيل إضافية في [Huitema 1998; Moy 1998; RFC 2328].



الشكل 4-40 تنظيم هرمي لنظام مستقل ذاتياً AS يتكون من أربع مناطق باستخدام بروتوكول OSPF.

المبادئ في الواقع العملي (Principles in Practice)

إعداد أوزان الوصلات في بروتوكول OSPF

افترضت مناقشتنا لتوجيه حالة الوصلة ضمنياً تهيئة أوزان الوصلة، وتشغيل خوارزمية توجيه مثل OSPF، وكذلك توجيه مرور البيانات طبقاً لجداول التوجيه المحسوبة بخوارزمية LS. من منظور السبب والتأثير، يتم إدخال أوزان الوصلات (بمعنى آخر تأتي بالمرتبة الأولى) وتنتج (عن طريق خوارزمية Dijkstra) مسارات التوجيه التي تقلّل الكلفة الإجمالية. من وجهة النظر هذه تعكس أوزان الوصلات كلفة استعمالها ويؤدي استخدام خوارزمية Disjkstra إلى تقليل الكلفة الإجمالية (وكمثال على ذلك إذا كان وزن الوصلة يتناسب عكسياً مع سعتها، فسيكون للوصلات ذات السعة العالية أوزان أصغر وبالتالي تصبح أكثر جاذبية من وجهة نظر التوجيه).

في الواقع العملي قد تُعكس علاقة السبب والتأثير بين أوزان الوصلات ومسارات التوجيه؛ بمعنى أنه قد يهيئ مشغلو الشبكة أوزان الوصلات للحصول على مسارات توجيه تحقق أهدافاً معينة لهندسة حركة المرور [Fortz 2000; Fortz 2002] كتوزيع الأحمال بشكل أفضل. افترض على سبيل المثال أن مشغل الشبكة يتوافر لديه تقديرٌ ما لتدفق حركة مرور البيانات التي تدخل الشبكة في كل نقطة دخول (ingress point) متجهة إلى كل نقطة خروج (egress point). عندئذٍ قد يريد المشغل تنفيذ خطة توجيه بعينها للتدفقات من نقاط الدخول إلى نقاط الخروج بحيث تقلل الاستخدام الأقصى للانتفاع بكل وصلات الشبكة. لكن مع خوارزمية توجيه مثل OSPF تُعتبر أوزان الوصلات بمثابة مقابض التحكم الرئيسية التي يقوم المشغل عن طريقها بضبط توجيه التدفق خلال الشبكة. وهكذا فللوصول لهدف تقليل الاستغلال الأقصى للوصلات لتوزيع أفضل للأحمال، يجب أن يوجِد مُشغِّل الشبكة مجموعة أوزان الوصلات التي تحقق هذا الهدف. هذا عكس علاقة السبب والتأثير، حيث يكون توجيه التدفق معروفاً، ويكون المطلوب إيجاد أوزان الوصلات بحيث تؤدي خوارزمية التوجيه إلى ذلك التوجيه المطلوب للتدفق.

4-6-3 التوجيه بين أنظمة AS : بروتوكول BGP

عرفنا الآن كيف يستخدم موفرو خدمة الإنترنت بروتوكولات RIP وOSPF لتحديد المسارات المثلى بين أزواج المصادر والوجهات التي تقع ضمن نظام AS بعينه. دعنا نتناول الآن كيفية تحديد المسارات بين أزواج المصدر والوجهة التي تمر عبر عدة أنظمة AS. تعد النسخة 4 من بروتوكول BGP والموصوفة في RFC 4271 المعيار الواقعي de facto (انظر أيضاً [RFC 1772; RFC 1773]) في الإنترنت اليوم، ومن

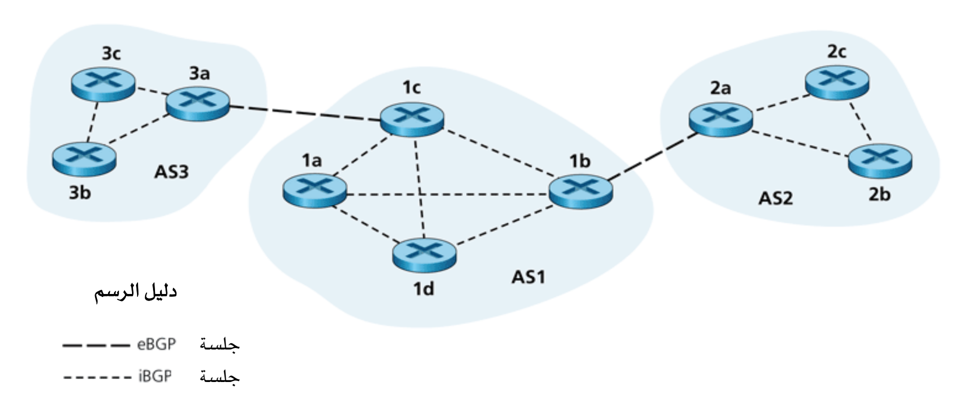
الشائع تسميته أيضاً باسم BGP4 أو ببساطة BGP. وكبروتوكول للتوجيه خارج AS (انظر الجزء 4-5-3)، فإنه يزوّد كل نظام AS بوسيلة لـ:

1. الحصول على معلومات الوصول للشبكات الفرعية من أنظمة AS المجاورة.
2. بث معلومات الوصول إلى كل الموجّهات داخل AS.
3. تحديد مسارات "جيدة" إلى الشبكات الفرعية طبقاً لمعلومات الوصول ولسياسة AS.

وبشكلٍ أكثر أهمية يسمح BGP لكل شبكة فرعية بالإعلان عن وجودها لبقيّة الإنترنت. تصرخ الشبكة الفرعية "أنا موجودة وأنا هنا"، ويتأكّد BGP من أن جميع أنظمة AS في الإنترنت تعلم بوجود الشبكة الفرعية وتعرف كيف تصل إليها. باختصار لولا بروتوكول BGP لأصبحت كل شبكة فرعية معزولة لوحدها ومجهولة لدى بقيّة الإنترنت.

أساسيات بروتوكول BGP

بروتوكول BGP معقّدٌ جداً؛ ولقد خصصت كتب بأكملها لهذا الموضوع وما زال العديد من القضايا غير مفهومة بشكلٍ واضح [Yannuzzi 2005]. علاوةً على ذلك وحتى بعد أن تقرأ الكتب وRFCs، قد تجد من الصعوبة أن تجيد BGP بالكامل بدون ممارسة لعدّة شهور (إن لم تكن سنوات) كمصمم أو كمشرف لشبكة موفر خدمة الإنترنت من الطبقات العليا. وعلى الرغم من ذلك ولكون BGP بروتوكولاً هاماً جداً للإنترنت (حيث يعتبر أساساً بمثابة الصمغ الذي يربط ما بين جميع أجزاء الشبكة)، فإننا نحتاج على الأقل لفهمٍ أولي لطريقة عمله. نبدأ بوصف كيفية عمل BGP ضمن سياق شبكة المثال البسيطة التي درسناها في وقت سابق في الشكل 4-32. في هذا الوصف نبني على مناقشتنا للتوجيه الهرمي في الجزء 4-5-3؛ وننصحك بمراجعة ذلك الموضوع.



الشكل 4-41 جلسات eBGP وiBGP.

في BGP تتبادل أزواج من الموجهات معلومات توجيه على توصيلات TCP شبه دائمة (semi-permanent) من خلال المنفذ 179. يبين الشكل 4-41 توصيلات TCP شبه الدائمة للشبكة الموجودة بالشكل 4-32. هناك عادةً توصيلة BGP TCP واحدة من هذا النوع لكل وصلة تربط مباشرة بين موجّهين في نظامي AS مختلفين؛ وهكذا ففي الشكل 4-41 توجد توصيلة TCP بين موجّهات البوّابة 3a و1c وتوصيلة TCP أخرى بين موجّهات البوّابة 1b و2a. هناك أيضاً توصيلات TCP نصف دائمة لبروتوكول BGP بين الموجّهات داخل نظام AS . وبشكلٍ محدد يعرض الشكل 4-41 إعداداً شائع الاستخدام يتضمن توصيلة TCP واحدة لكل زوج من الموجّهات التي بداخل AS مما يُشكِّل شبكة ربط من توصيلات TCP داخل كل AS. لكل توصيلة TCP يُطلق على الموجّهين في نهاية التوصيلة نظائر BGP، وتدعى توصيلة TCP مع كل رسائل BGP المرسلة عبرها "جلسة BGP". علاوةً على ذلك تُسمى جلسة BGP التي تغطّي اثنين من أنظمة AS جلسة BGP خارجية (eBGP)، وتُسمى جلسة BGP بين الموجهات في نفس AS جلسة BGP داخلية (iBGP). في الشكل 4-41 توضح جلسات eBGP بخطوط متقطعة بشُرَط طويلة؛ وجلسات iBGP بخطوط متقطعة بشُرَط قصيرة. لاحظ أن خطوط جلسات BGP في الشكل 4-41 لا تناظر بالضرورة الوصلات المادية في الشكل 4-32.

يسمح BGP لكل AS أن يحدد أي الوجهات يمكن الوصول إليها عن طريق أنظمة AS المجاورة له. في BGP هذه الوجهات ليست مضيفات ولكنها بادئات عناوين CIDR، حيث يمثل كلٌّ منها شبكة فرعية أو مجموعة شبكات فرعية. وهكذا فعلى سبيل المثال افترض أن هناك أربع شبكات فرعية متصلة بـ :AS2 138.16.64/24، 138.16.65/24، 138.16.66/24، 138.16.67/24. عندئذ يمكن أن يجمع AS2 البادئات لهذه الشبكات الفرعية الأربع ويستعمل BGP لإعلان بادئة واحدة 138.16.64/22 إلى AS1. وكمثال آخر افترض أن الشبكات الثلاث الأولى فقط من تلك الشبكات الفرعية الأربع موجودة في AS2 والشبكة الفرعية الرابعة 138.16.67/24 موجودة في AS3. وكما وصفنا في المبادئ والواقع العملي في الجزء 4-4-2، نظراً لأن الموجّهات تستخدم تطابق البادئة الأطول لتمرير رزم البيانات، يمكن أن يعلن AS3 لـ AS1 البادئة الأكثر تحديداً 138.16.67/24، ولايزال بوسع AS2 أن يعلن لـ AS1 البادئة المجمّعة 138.16.64/22.

دعنا نتناول الآن كيف يوزّع BGP معلومات الوصول للبادئات على جلسات BGP الموضحة في الشكل 4-41. كما قد تتوقّع، باستخدام جلسة eBGP بين موجهات البوّابة 3a و1c، يرسل AS3 إلى AS1 قائمة بالبادئات التي يمكن الوصول إليها من AS3؛ ويرسل AS1 إلى AS3 قائمة بالبادئات التي يمكن الوصول إليها من AS1. بنفس الطريقة يتبادل AS1 وAS2 معلومات الوصول خلال موجّهات البوّابة 1b و2a. أيضاً كما قد تتوقّع عندما يستلم موجّه بوّابة (في أي نظام AS) بادئات تم معرفتها عن طريق eBGP، يستخدم موجّه البوّابة جلسات iBGP لتوزيع البادئات إلى الموجّهات الأخرى في AS. وهكذا تتعرف كل الموجّهات في AS1 على بادئات AS3 بما في ذلك موجّه البوّابة 1b. يمكن أن يعيد موجّه البوّابة 1b (في AS1) إعلان بادئات AS3 إلى AS2. عندما يعرف موجّه (بوّابة أو غيره) بادئة جديدة فإنه ينشئ مدخلاً للبادئة في جدول التمرير لديه كما وصفنا في الجزء 4-5-3.

خواص المسار ومسارات BGP

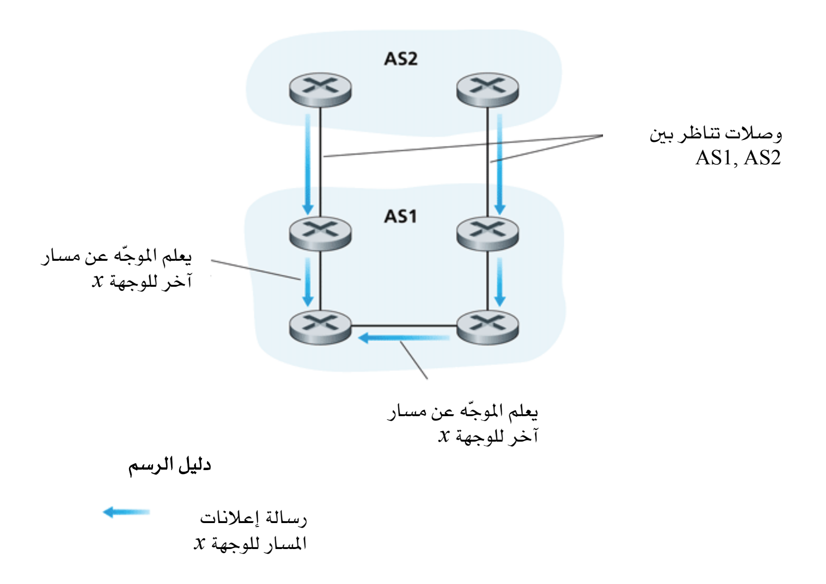
بعد هذا الفهم التمهيدي لـ BGP دعنا نغوص فيه بعمق بعض الشيء (مع مواصلة غض الطرف عن بعض التفاصيل الأقل أهمية). في BGP، يتم تمييز نظام مستقل ذاتياً (AS) برقم فريد عالمياً ((Autonomous System Number (ASN)) [RFC 1930]. من الناحية الفنية ليس كل AS له رقم ASN. وبالتحديد فإن النظام الذي يطلق عليه نظام عقب stub AS (والذي يحمل فقط حركة مرور البيانات التي يكون هو مصدرها أو وجهتها) لن يكون له عادةً رقم ASN؛ وسنهمل هذا التفصيل في مناقشتنا لكي نتمكن من رؤية الغابة بدلاً من الأشجار. يتم تخصيص أرقام AS، مثلها في ذلك مثل عناوين IP، عن طريق مكاتب تسجيل ICANN الإقليمية [ICANN 2007].

عندما يعلن موجّه بادئة عبر جلسة BGP، يُضمّن مع البادئة عدداً من بارامترات BGP. في مفردات BGP يطلق على البادئة مع البارامترات الخاصة بها اسم "مسار" (route). وهكذا تعلن نظائر BGP المسارات لبعضها البعض. تُعتبر AS-PATH وNEXT-HOP من البارامترات الأكثر أهمية:

- بارامتر AS-PATH: يتضمن هذا البارامتر أنظمة AS التي مر عبرها إعلان البادئة. عندما تمر بادئة عبر نظام AS يضيف AS رقم ASN الخاص به إلى البارامتر AS-PATH. على سبيل المثال بالنظر إلى الشكل 4-41 وبافترض أن البادئة 138.16.64/24 أُعلنت أولاً من AS2 إلى AS1. إذا أعلن AS1 بعدئذٍ البادئة إلى AS3، فسيتضمن البارامتر AS-PATH AS2 AS1. تستخدم الموجّهات ذلك البارامتر لاكتشاف ومنع دوران الإعلانات في حلقات مفرغة؛ وبالتحديد إذا رأى موجّه أن نظام AS التي ينتمي له موجودُ ضمن بارامتر AS-PATH فسيرفض الإعلان. كما سنناقش قريباً تستخدم الموجّهات AS-PATH أيضاً في الاختيار ما بين المسارات المتعدّدة إلى نفس البادئة.
- بارامتر NEXT-HOP: يلعب هذا البارامتر دوراً مهماً في الربط ما بين بروتوكولات التوجيه داخل نظام AS وبروتوكولات التوجيه خارجه. تمثل NEXT-HOP واجهة الموجّه الموجود في بداية AS-PATH. ولفهم هذا

البارامتر، دعنا نشير ثانية للشكل 4-41. لننظر ما يحدث عندما يعلن موجه البوّابة 3a الموجود في AS3 مساراً إلى موجّه البوّابة lc في AS1 باستخدام eBGP. يتضمّن المسار البادئة المعلنة (والتي سنسميها x) والبارامتر AS-PATH للبادئة. يتضمّن هذا الإعلان أيضاً NEXT-HOP والذي يمثل عنوان IP لواجهة الموجّه a3 التي توصل إلى c1. (تذكّر أن الموجّه له عدّة عناوين IP؛ واحد لكل واجهة من واجهاته) انظر الآن لما يحدث عندما يعرف الموجّه ld عن المسار من iBGP. بعد العلم عن هذا المسار إلى x، قد يريد الموجّه ld إرسال الرزم إلى x على طول المسار، أي قد يريد الموجّه ld تضمين المدخل (x, l) في جدول التمرير لديه حيث تمثل l واجهته التي تبدأ المسار الأدنى كلفة من ld نحو موجّه البوّابة lc. ولتحديد l يزوّد 1d عنوان IP في خاصية NEXT-HOP للتوجيه داخل AS. لاحظ أن خوارزمية التوجيه داخل AS تحدد المسار الأدنى كلفةً إلى كل الشبكات الفرعية المتصلة بالموجّهات في AS1، ويتضمن ذلك الشبكة الفرعية للوصلة بين 1c و3a. من هذا المسار الأدنى كلفةً من ld إلى الشبكة الفرعية lc-3a يحدد ld واجهته l التي تقع على بداية هذا المسار وبعد ذلك يضيف المدخل (x, l) إلى جدول التمرير لديه. الخلاصة: إنّ الموجّهات تستخدم البارامتر AS-PATH لتشكيل جداول التمرير لديها بشكلٍ صحيح.

يوضح الشكل 4-42 حالة أخرى نحتاج فيها إلى البارامتر AS-PATH. في هذا الشكل تتصل AS1 وAS2 بوصلتي نظير. يمكن أن يعرف موجّه في AS1 مسارين مختلفين إلى نفس البادئة x. يمكن أن يكون لهذين المسارين نفس البارامتر AS-PATH إلى x، لكن يمكن أن يكون لهما قيماً مختلفة للبارامتر NEXT-HOP تقابل وصلات النظير المختلفة. باستعمال قيم AS-PATH وخوارزمية التوجيه داخل AS، يمكن أن يحدد الموجّه كلفة المسار إلى كل وصلة نظير، وبعد ذلك يطبّق توجيه البطاطس الساخنة (انظر الجزء 4-5-3) لتحديد الواجهة الملائمة.



الشكل 4-42 تُستخدم بارامترات NEXT-HOP المتضمنة في الإعلانات لتحديد أي وصلة نظير سيتم استخدامها.

يتضمّن بروتوكول BGP أيضاً بارامترات تسمح للموجّهات بتخصيص معايير مفاضلة للمسارات وبارامتر يبين كيف تم إدخال البادئة إلى بروتوكول BGP في نظام AS المصدري. لمناقشة كاملة عن خواص المسارات اطّلِع على [Griffin 2002; Stewart 1999; Halabi 2000; Feamster 2004; RFC 4271].

عندما يستلم موجّه بوّابة إعلاناً من موجّه آخر فإنه يستعمل سياسته للاستيراد (import policy) لتقرير ما إذا كان سيقبل أو يستبعد المسار، وما إذا كان سيضع بعض البارامترات مثل معايير المفاضلة. قد تستبعد سياسة الاستيراد مساراً لأن AS لا يريد مرور البيانات على أحد أنظمة AS الموجودة في بارامتر AS-PATH للمسار. قد يستبعد موجّه البوّابة المسار أيضاً لأنه يعرف مساراً مفضّلاً إلى نفس البادئة.

اختيار مسار BGP

كما وصفنا في وقت سابق في هذا الجزء يستخدم BGP البروتوكول eBGP وiBGP لتوزيع المسارات إلى كل الموجّهات ضمن أنظمة AS. من هذا التوزيع قد

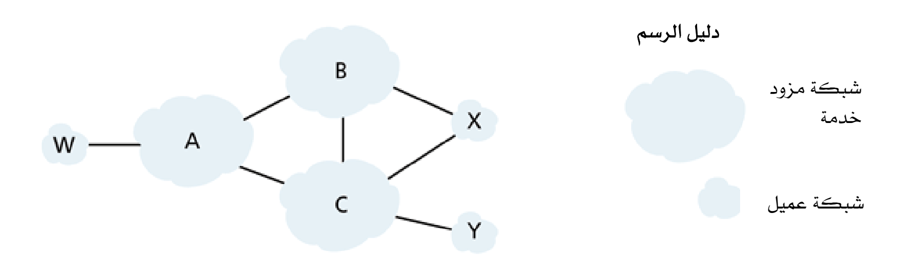
يتعرف موجّه على أكثر من مسار لبادئة معينة، في هذه الحالة يجب أن يختار الموجّه أحد تلك المسارات المحتملة. تشمل المدخلات لعملية اختيار المسار هذه مجموعة المسارات التي تم التعرف عليها وقبولها بالموجّه. في حالة وجود مسارين أو أكثر إلى نفس البادئة، عندئذ يطبق BGP قواعد الحذف التالية بشكلٍ متسلسل إلى أن يبقى مسار واحد:

- تخصص للمسارات قيم مفاضلة محلية كأحد البارامترات الخاصة بها. قد يتم ضبط المفاضلة المحلية للمسار من قِبَل الموجّه نفسه أو تكون مما تعلّمه موجّه آخر في نفس نظام AS. هذا القرار يتعلق بسياسة التشغيل ويترك لمشرف شبكة AS. (سنناقش قضايا سياسة BGP بعد قليل بشيء من التفصيل). يتم اختيار المسارات ذات قيم المفاضلة المحلية الأعلى.
- من المسارات الباقية (والتي لها جميعاً نفس قيم المفاضلة المحلية)، يتم اختيار المسار الذي له أقصر AS-PATH. إذا كانت تلك هي القاعدة الوحيدة لاختيار المسار، فإن BGP يكون في الواقع مستخدماً لخوارزمية DV لتحديد المسار، حيث يستخدم معيار المسافة عدد القفزات عبر أنظمة AS بدلاً من عدد القفزات عبر موجّهات.
- من بين المسارات المتبقية (والتي لها جميعاً نفس قيم المفاضلة المحلية ونفس طول AS-PATH)، يتم اختيار المسار الذي له أقرب NEXT-HOP. ونعني بـ "أقرب" هنا الموجّه بمسار له أصغر قيمة لأدنى كلفة حسب ما تحدده خوارزمية التوجيه داخل AS. كما نوقش في الجزء 4-5-3 تدعى هذه العملية توجيه البطاطس الساخنة.
- إذا تبقى أكثر من مسار واحد يستخدم الموجّه معرّفات BGP لاختيار المسار؛ انظر [Stewart 1999].

إن قواعد الحذف في الواقع أكثر تعقّيداً من تلك الموصوفة هنا. ولتفادي الكوابيس حول BGP من الأفضل أن نتعلم قواعد الاختيار في BGP بجرعات صغيرة!

سياسة التوجيه

دعنا نوضّح بعض المفاهيم الأساسية لسياسة التوجيه في BGP بمثال بسيط. يوضح الشكل 4-43 ستّة أنظمة مستقلة ذاتياً متصلة ببعضها: A، B، C، W، X، Y. من المهم ملاحظة أن A، B، C، W، X، Y أنظمة مستقلة ذاتياً وليست موجّهات. دعنا نفترض أن الأنظمة المستقلة ذاتياً W، X، Y شبكات عقب (stub) وأن A، B، C شبكات عمود فقري لموفر خدمة. سنفترض أيضاً أن A، B، C لها جمعياً وصلة نظير مع بعضها البعض، كما توفر معلومات BGP كاملةً إلى شبكات عملائها. يجب أن يكون كل المرور الداخل لشبكة عقب (stub) متّجه إلى تلك الشبكة، وأن يكون كل المرور الخارج من شبكة stub متولداً في تلك الشبكة. واضح أن W وY تُعتبر شبكات عقب. علاوةً على ذلك تُعتبر X شبكة عقب متعددة المنازل (multihomed) حيث توصّل إلى بقيّة الشبكة عن طريق موفري خدمة مختلفين (وهي طريقة أصبحت شائعة على نحو متزايد عملياً). ومع ذلك فمثلها مثل W وY يجب أن تكون X نفسها هي المصدر/الوجهة لكل المرور الخارج/الداخل لها.



الشكل 4-43 سيناريو BGP بسيط.

لكن كيف يمكن تطبيق وفرض هذا السلوك على شبكة العقب؟ كيف سيتم منع X من تمرير حركة المرور بين B وC؟ يمكن تحقيق ذلك بسهولة بالتحكم في الكيفية التي تعلن بها مسارات BGP. وبالتحديد ستعمل X كشبكة عقب إذا أعلنت (لجيرانها B وC) بأنها ليس لها مسارات إلى أي وجهات أخرى ماعدا

نفسها. أي أنه رغم كون X قد تعرف مساراً (مثلاً XCY والذي يصل إلى Y) إلا أن X لن تعلن عن هذا المسار لـ B. ولأن B لا تدري أن X عندها مسار إلى Y فإن B لن ترسل أبداً حركة بيانات متجهة إلى Y (أو C) عن طريق X. هذا المثال البسيط يوضح كيف يمكن أن تستخدم سياسة إعلان انتقائية للمسارات في تطبيق علاقات التوجيه بين العملاء وموفري الخدمة.

دعنا نركز على شبكة موفِّر خدمة، مثلاً نظام B، وافترض أنه عرف (من A) أن A لديه مسار AW إلى W. يمكن أن يضيف B المسار BAW إلى قاعدة معلومات التوجيه لديه. من الواضح أن B يريد أيضاً أن يعلن المسار BAW إلى زبونه X لكي يعرف X أنه بوسعه التوجيه لـ W عن طريق B. لكن هل يجب أن يعلن B المسار BAW إلى C؟ إذا فعل ذلك يمكن أن يوجّه C المرور إلى W عن طريق CBAW. إذا كان كلٌّ من A، B، C شبكات عمود فقري عندئذ قد يشعر B — ومعه حق — بأنه لا يجب أن يتحمل عبئاً (وكلفة!) نقل حركة المرور بين A وC. قد يكون B محقاً في شعوره بأن هذه مهمة (وكلفة!) A وC في التأكد من أن C يمكنه التوجيه إلى زبائن A ومنهم عبر وصلة مباشرة بين A وC. لا توجد حالياً معايير رسمية تحكم كيفية التوجيه بين شبكات العمود الفقري لموفري خدمة الإنترنت. ومع ذلك فالطريقة المجرّبة والمتبعة من قِبَل موفري خدمة الإنترنت التجاريين هي أن أي حركة مرور تتدفّق عبر شبكة عمود فقري لموفر خدمة الإنترنت يجب أن يكون لها إمّا مصدر أو وجهة (أو كلاهما) في شبكة تُعتبر زبوناً لموفِّر خدمة الإنترنت هذا؛ وإلا فإن حركة المرور ستحصل على "توصيلة مجانية" عبر شبكة موفر الخدمة. يتم التفاوض عادة على اتفاقيات ثنائية بين أزواج موفري خدمة الإنترنت (والتي تتناول تساؤلات كالمذكورة أعلاه)، وتكون تلك الاتفاقيات سرّية في أغلب الأحيان. يتضمن [Huston 1999a] مناقشة ممتعة للاتفاقيات بين النظائر. ولوصف مفصّل للكيفية التي تعكس بها سياسة التوجيه العلاقات التجارية بين موفري خدمة الإنترنت، راجع [Gao 2001]. لمناقشة حديثة لسياسات توجيه BGP من وجهة نظر موفر خدمة الإنترنت، اطّلِع على [Caesar 2005].

المبادئ في الواقع العملي (Principles in Practice)

لماذا توجد بروتوكولات مختلفة للتوجيه داخل أنظمة AS وفيما بينها؟

بعد أن درسنا الآن تفاصيل بروتوكولات محددة للتوجيه داخل أنظمة AS وفيما بينها في إنترنت اليوم، دعنا نختم بمحاولة للإجابة على سؤال قد يكون هو الأكثر أهمية والذي يمكن أن نسأله في المقام الأول عن تلك البروتوكولات (نأمل أن تكون قد تساءلت عن هذا دوماً ولم تغب عنك الغابة بالنظر للأشجار!): لماذا تستخدم بروتوكولات مختلفة للتوجيه داخل أنظمة AS وفيما بينها؟

تكمن الإجابة على هذا السؤال في الاختلافات الأساسية بين أهداف التوجيه داخل أنظمة AS وفيما بينها:

- سياسات التشغيل: تلعب سياسات التشغيل دوراً مهماً في عمليات التوجيه بين أنظمة AS. ربّما يكون من المهم منع حركة المرور التي تنشأ في AS معين من عبور AS آخر. وبالمثل قد يريد AS معين أن يسيطر على حركة المرور التي تعبره إلى أنظمة AS الأخرى. رأينا أن بروتوكول BGP ينقل بارامترات المسارات ويسمح بالتحكم في توزيع معلومات التوجيه ليتسنى اتخاذ قرارات التوجيه المعتمدة على سياسة معينة. أما داخل نظام AS فيعتبر كل شيء اسمياً تحت نفس الرقابة الإدارية وبالتالي تلعب السياسة دوراً أقل أهمية بكثير في اختيار المسارات داخل AS.
- القدرة على التوسع: تُعتبر قدرة خوارزمية التوجيه وهياكل بياناتها على معالجة التوجيه إلى/بين أعداد كبيرة من الشبكات قضية حاسمة في التوجيه بين أنظمة AS، في حين تُعتبر القدرة على التوسع داخل AS أقل أهميةً. وأحد أسباب ذلك أنه إذا زاد حجم نطاق إداري جداً فيمكن دوماً تقسيمه إلى عدة أنظمة AS والقيام بعملية التوجيه بينها. (تذكّر أن بروتوكول OSPF يسمح بتكوين مثل هذا التدرج الهرمي بتقسيم AS إلى مناطق).
- الأداء: نظراً لأن التوجيه بين أنظمة AS يعتمد كثيراً على السياسة فإن معايير الجودة للمسارات المستعملة (كالأداء مثلاً) تُشكّل في أغلب الأحيان اهتماماً ثانوياً (بمعنى أنه قد يُفضّل مسارٌ أطول أو أكثر كلفة على مسار أقصر وأقل كلفة إذا كان الأول يحقق بعض معايير السياسة التي لا يحققها الثاني). في الحقيقة كما رأينا في التوجيه بين أنظمة AS ليس هناك ذكر للكلفة مرتبط بالمسارات (باستثناء عدد قفزات AS). في المقابل داخل نظام AS تكون مثل هذه المخاوف السياسية أقل أهميةً مما يسمح للتوجيه بالتركيز أكثر على مستوى الأداء الذي يمكن تحقيقه على المسار.

كما ذكرنا سابقاً يُعدّ BGP معياراً واقعياً (de facto standard) للتوجيه بين أنظمة AS في الإنترنت العامّة. ولرؤية محتويات جداول التوجيه المختلفة (والكبيرة!) لبروتوكول BGP والمستخلصة من موجّهات موفري خدمة الإنترنت في الطبقة الأولى (tier-1) راجع الموقع http://www.routeviews.org. في أغلب الأحيان تحتوي جداول توجيه BGP على عشرات الآلاف من البادئات والبارامترات المناظرة. توجد إحصائيات حول حجم وخصائص جداول توجيه BGP في [Huston 2001; Meng 2005]. وبهذا تكتمل مقدمتنا القصيرة عن بروتوكول BGP. إن فهم BGP مهم لأنه يلعب دوراً مركزياً في الإنترنت. ونشجّعك على الاطلاع على المراجع [Griffin 2002; Stewart 1999; Labovitz 1997; Halabi 2000; Huitema 1998; Gao 2001; Feamster 2004; Caesar 2005] للمزيد من المعلومات حول BGP.

7-4 التوجيه الإذاعي والمتعدّد (Broadcast and multicast routing)

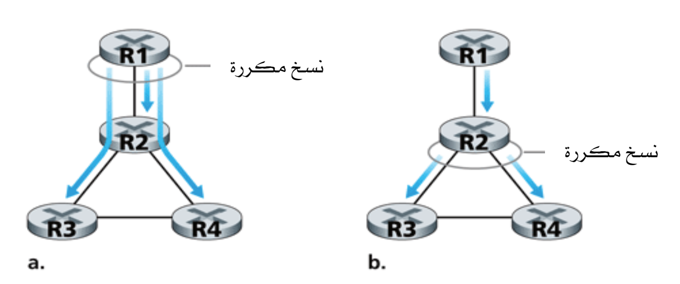
ركّزنا في هذا الفصل حتى الآن على بروتوكولات التوجيه التي تدعم الاتصال الفردي (unicast) (أي من نقطة إلى نقطة)، والذي فيه ترسل عقدة مصدر وحيدة الرزمة إلى عقدة وجهة وحيدة. في هذا الجزء سنحوّل انتباهنا لبروتوكولات توجيه الإذاعة والتوجيه المتعدد. في توجيه الإذاعة توفر طبقة الشبكة خدمة تسليم رزمة أرسلت من عقدة مصدر إلى كل العقد الأخرى في الشبكة؛ أما في التوجيه المتعدّد يمكّن لعقدة مصدر وحيدة إرسال نسخة من رزمة إلى مجموعة جزئية من عقد الشبكة الأخرى. في الجزء 4-7-1 سنناقش خوارزميات توجيه الإذاعة وتضمينها في بروتوكولات التوجيه، ثم نفحص التوجيه المتعدّد في الجزء 4-7-2.

4-7-1 خوارزميات توجيه الإذاعة

قد تكون أبسط طريقة لتحقيق اتصال إذاعي هي أن ترسل عقدة المصدر نسخة مستقلة من الرزمة إلى كل وجهة ممكنة (باستخدام توجيه إرسال أحادي unicast)، كما هو موضح في الشكل 4-44 (a). فإذا كان عدد عقد الوجهات N فإن عقدة المصدر ببساطة تُعد N نسخة من الرزمة؛ وتعنون كل نسخة منها إلى أحد الوجهات المختلفة ثم ترسلها إلى تلك الوجهات. هذه طريقة بسيطة للإذاعة (أي ما

يعرف بـ *N*-way unicast) ، ولا تحتاج إلى بروتوكولات جديدة لطبقة الشبكة ولا إلى إجراءات إضافية لنسخ الرزمة أو التمرير). ومع ذلك هناك عدّة عيوب لهذه الطريقة. العيب الأول عدم كفاءتها. إذا كانت عقدة المصدر موصّلة إلى بقيّة الشبكة عن طريق وصلة وحيدة، فإن عدد *N* نسخة من نفس الرزمة ستعبر هذه الوصلة الوحيدة. واضحٌ أن العملية ستكون أكثر كفاءة إذا أرسلنا نسخة واحدة فقط من الرزمة على تلك القفزة الأولى على أن تقوم العقدة على الطرف الآخر من تلك الوصلة بإرسال أي نسخ إضافية مطلوبة. أي سيكون الحل أكثر كفاءة إذا جعلنا عقد الشبكة نفسها (بدلاً من عقدة المصدر فقط) تنشئ نسخاً مضاعفة من الرزمة. على سبيل المثال في الشكل 4-44 (b) تعبر نسخة واحدة فقط الوصلة R1-R2. ثمّ تنسخ تلك الرزمة في R2 وترسل نسخة على كلٍّ من الوصلات R2-R3 وR2-R4.

إن العيوب الأخرى لطريقة *N*-way unicast ربما تكون أكثر دقة ولكنها ليست أقل أهمية. إن طريقة *N*-way unicast تفترض ضمناً أن عقد الاستقبال وعناوينها معروفة للمُرسِل. لكن كيف يحصل المُرسِل على تلك المعلومات؟ على الأغلب سنحتاج إلى آليات بروتوكولات إضافية (كبروتوكول عضوية الإذاعة أو بروتوكول التسجيل للوجهة). سيضيف هذا عبئاً إضافياً وبشكلٍ هام تعقيداً إضافياً للبروتوكول الذي بدا بسيطاً جداً في البداية. وعيب أخير لطريقة N-way unicast يتعلق بالأغراض التي من أجلها سيستخدم الإرسال الإذاعي. في الجزء 4-5 عرفنا أن بروتوكولات توجيه حالة الوصلة تستخدم الإرسال الإذاعي لنشر معلومات حالة الوصلة والتي تُستخدم لحساب مسارات الإرسال الفردي unicast. واضح أنه حين يُستخدم الإرسال الإذاعي في حساب وتحديث مسارات unicast يكون من غير المعقّول (في أحسن الأحوال!) الاعتماد على البنية التحتية لتوجيه unicast لإنجاز الإرسال الإذاعي.



الشكل 4-44 نسخ الرزمة عند المصدر في مقابل نسخها في الشبكة.

من هذه العيوب العديدة لطريقة *N*-way unicast تتضح أهمية الطرق التي تلعب فيها عقد الشبكة نفسها دوراً فاعلاً في نسخ الرزمة وتمريرها وحساب مسارات الإذاعة. سنتناول فيما يلي عدداً من تلك الطرق ونستخدم مرة أخرى اصطلاحات الرسم البياني التي قدّمناها في الجزء 4-5. نمثل الشبكة مرة أخرى كرسم بياني $G=(N, E)$ حيث N مجموعة العقد وE مجموعة الحافات، وحيث كل حافة هي زوج من العقد N. سنكون متساهلين نوعاً ما في استخدامنا للرموز وعندما لا يكون هناك مجال للخلط في الفهم سنستخدم N للإشارة إلى مجموعة العقد وأيضاً إلى حجم تلك المجموعة ($|N|$).

الفيض غير المحكوم

تُعتبر طريقة الفيض (flooding) أبسط الطرق لإنجاز الإرسال الإذاعي، حيث ترسل عقدة المصدر نسخة الرزمة إلى كل جيرانها. عندما تستلم عقدةٌ ما الرزمة المذاعة، فإنها تنسخ الرزمة وترسلها إلى كل جيرانها (ماعدا الجار الذي استلمت منه الرزمة). واضح أنه إذا كان الرسم البياني للشبكة متصلاً، فستؤدي هذه الطريقة إلى أن تتلقى كل عقدة في الرسم البياني في النهاية نسخةً من الرزمة المذاعة. وبالرغم من أن هذه الطريقة بسيطة ورائعة، إلا أن لها عيب قاتل (قبل أن تستمر في القراءة، فكِّر لترى إذا كان يمكنك أن تفهم هذا العيب القاتل):

في حالة وجود حلقات مفرغة بالرسم البياني للشبكة، فعندئذ ستستمر نسخة أو أكثر من كل رزمة مذاعة في الدوران بشكلٍ غير محدود. على سبيل المثال في الشكل 4-44 ستفيض R2 على R3، وتفيض R3 على R4، وتفيض R4 على R2، ومرة أخرى تفيض R2 على R3، وهكذا. هذا السيناريو البسيط يؤدّي إلى دوران لانهائي لرزمتى إذاعة، أحداهما باتجاه دوران عقارب الساعة، والآخر بعكس اتجاه دوران عقارب الساعة.

ولكن يمكن أيضاً أن يكون هناك عيب آخر قاتل ومفجع بدرجة أكبر: عندما توصّل عقدة إلى أكثر من عقدتين أخريين، ستنشئ وترسل عدة نسخٍ من الرزمة المذاعة، وستقوم كلٌّ منها بدورها لإنشاء عدة نسخ (في العقد الأخرى التي لها أكثر من جارين)، وهكذا. سيكون نتيجة ذلك عاصفة من الرزم المذاعة والتي تقود في النهاية إلى جعل الشبكة عديمة الفائدة. (انظر إلى تمارين الواجب المنزلي في نهاية الفصل لإحدى المسائل التي يتم فيها تحليل المعدل الذي تنمو به مثل تلك العاصفة).

الفيض المحكوم

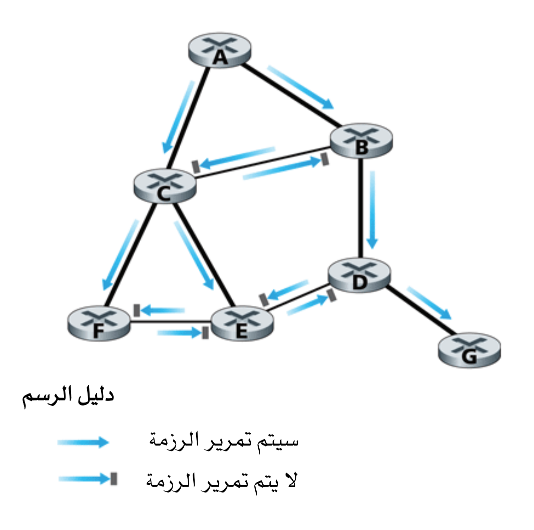
يكمن المفتاح لتجنب عاصفة الإذاعة تلك في جعل العقدة تختار بتعقّل متى تفيض برزمة ومتى تُحجم عن ذلك (مثلاً إذا سبق أن استلمت نسخة من نفس الرزمة وفاضت بها). عملياً يمكن أن يتم ذلك بعدّة طرق.

في الفيض المحكوم بالرقم المتسلسل تضيف عقدة المصدر عنوانها (أو معرّفاً فريداً آخر) بالإضافة إلى رقم متسلسل لرزمة الإذاعة، ثمّ ترسل الرزمة إلى كل جيرانها. تحتفظ كل عقدة بقائمة من عناوين المصدر الرقم المتسلسل لكل رزمة مذاعة تستلمها وتنسخها وترسلها. عندما تستلم عقدة رزمة مذاعة سوف تفحص القائمة أولاً لترى ما إذا كانت الرزمة موجودة بالقائمة. فإذا حدث ذلك فإنها تقوم بإسقاط الرزمة؛ وإلا فسوف تنسخها وترسلها لكل جيرانها (ماعدا العقدة التي استلمت منها الرزمة). يستخدم بروتوكول النظائر Gnutella (والذي ناقشناه في الفصل الثاني) الفيض المحكوم بالأرقام المتسلسلة لإذاعة الاستفسارات في الشبكة

الإضافية (overlay network). (إلا أن نسخ الرسائل وتمريرها في Gnutella يتم في طبقة التطبيقات بدلاً من طبقة الشبكة).

من الطرق الأخرى للتحكم في الفيض طريقة تعرف بتمرير المسار العكسي (Reverse Path Forwarding (RPF)) [Dalal 1978]، كما تُدعى أحياناً باسم إذاعة المسار العكسي (RPB). إن الفكرة وراء RPF بسيطة ورائعة. عندما يستلم موجّه رزمةً مذاعةَ بعنوان مصدر معين فإنه يرسلها على كل وصلاته الخارجة (ماعدا تلك التي استلم منها الرزمة) ويتم هذا فقط إذا وصلت الرزمة على الوصلة الموجودة على أقصر مسار unicast من الموجّه إلى المصدر. فيما عدا ذلك يهمل الموجّه ببساطة الرزمة القادمة بدون تمريرها على أيٍّ من وصلاته الخارجة. يمكن إسقاط مثل تلك الرزمة لأن الموجّه يعرف أنه سوف يستلم أو قد استلم نسخة من تلك الرزمة على الوصلة التي على طريقه الأقصر الذي يعود إلى المُرسِل. (قد تريد إقناع نفسك بأن هذا سيحدث في الواقع وأنه لن يكون هناك مجال لحدوث دوران الرزم أوعاصفة الإرسال الإذاعي). لاحظ أن RPF في الحقيقة لا يستعمل توجيه unicast لتسليم رزمة إلى وجهة ما، ولا يتطلّب أن يعرف الموجّه المسار الأقصر بكامله بينه وبين المصدر. يحتاج RPF فقط لمعرفة الجار التالي على مسار unicast الأقصر إلى المُرسِل؛ حيث يستخدم هوية هذا الجار فقط لتحديد ما إذا كان سيرسل فيضاً من رزمة الإذاعة التي استلمها أم لا.

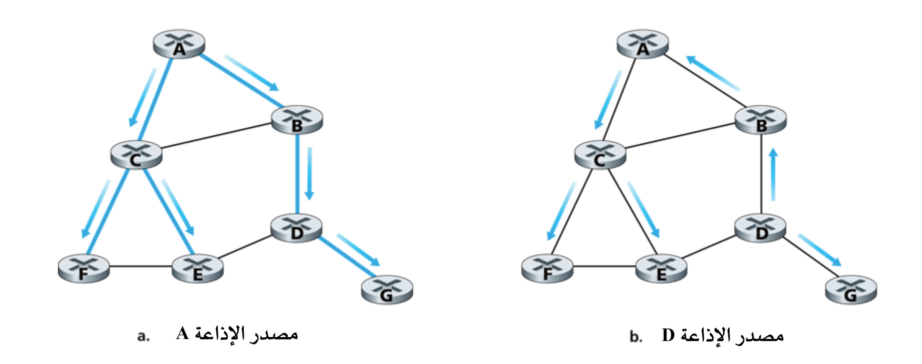
يوضح الشكل 4-45 بروتوكول RPF. افترض أن الوصلات المرسومة بخطوط سميكة تمثل مسارات أدنى كلفة من المستلمين إلى المصدر (A). تذيع العقدة A في البداية الرزمة التي مصدرها A إلى العقد C وB. ترسل العقدة B رزمة المصدر A التي استلمتها من A (لأن A على مسارأدنى كلفة من B إلى A) إلى كل من C وD. ستهمل B (تُسقط بدون تمرير) أي رزمة من المصدر A تتلقاها من أي عقدة أخرى (مثلاً C أو D). دعنا الآن نأخذ في الاعتبار عقدةً ولتكن C والتي تستلم رزمة من المصدر A مباشرة من A وكذلك من B. لأن B ليست على مسار C الأقصر الذي يعود إلى A، فإن C ستهمل أي رزمة من المصدر A تتلقاها من B. من ناحية أخرى عندما تستلم C رزمة من المصدر A مباشرة من A، فسترسل الرزمة إلى العقد B، E، F.



الشكل 4-45 تمرير المسار العكسي.

الإذاعة عبر الشجرة الممتدة (Spanning Tree Broadcast)

بينما يتفادى الفيض المحكوم بالأرقام المتسلسلة عاصفة الإذاعة، فإنه لا يتفادى بشكلٍ كامل إرسال رزم مذاعة غير ضرورية (زائدة عن الحاجة). على سبيل المثال في الشكل 4-46 تستلم العقد B، C، D، E، F رزمة أو رزمتين غير ضروريتين. وبشكلٍ مثالي يجب أن تستلم كل عقدة نسخة واحدة فقط من الرزمة المذاعة. بفحص الشجرة المكونة من العقد الموصلة بالخطوط السميكة في الشكل 4-46 (a)، يمكنك ملاحظة أنه إذا أُرسلت الرزم المذاعة فقط على طول الوصلات ضمن هذه الشجرة، فإن كل عقدة بالشبكة ستستلم نسخة واحدة فقط من الرزمة المذاعة (هذا بالضبط هو الحل الذي كنا نبحث عنه!). هذه الشجرة مثال للشجرة الممتدة (spanning tree)، وهي شجرة تحتوي كل العقد في الرسم البياني. وبشكلٍ أكثر رسمية الشجرة الممتدة للرسم البياني $G=(N, E)$ هي رسم بياني $G'=(N, E')$ حيث تمثل E' مجموعة جزئية من E، كما أن G' رسم بياني متصل يحتوي على كل العقد الأصلية في G ولا يحتوي على حلقات (cycles).



الشكل 4-46 البث الإذاعي عبر شجرة ممتدة.

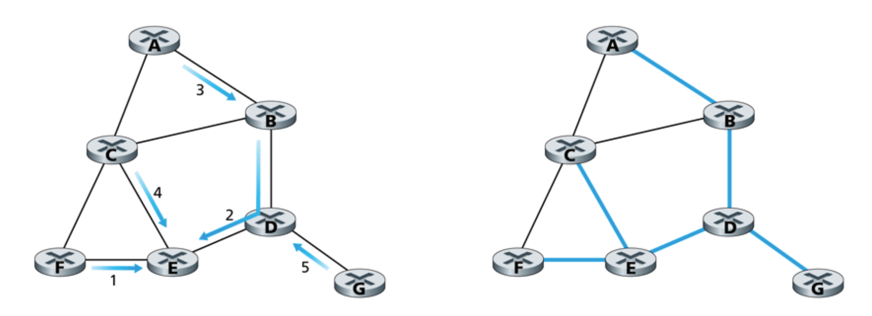
إذا كان لكل وصلة كلفة مصاحبة فإن كلفة الشجرة تساوي مجموع كلف وصلاتها، وعندئذ (بما لا يدعو للاستغراب) يطلق على الشجرة التي لها أدنى كلفة بين كل الأشجار الممتدة عبر الرسم البياني "الشجرة الممتدة بأدنى كلفة" (minimum-spanning tree).

ومن ثم فإن إحدى الطرق الأخرى للإرسال الإذاعي تتلخص في أن تبني عقد الشبكة أولاً شجرة ممتدة. عندما تريد عقدة المصدر أن تذيع رزمة، فإنها ترسل الرزمة على كل وصلاتها التي تنتمي للشجرة الممتدة. إذا استلمت عقدة رزمة مذاعة ترسل الرزمة إلى كل جيرانها في الشجرة الممتدة (ماعدا الجار الذي استلمت منه الرزمة). لا تتخلص الشجرة الممتدة فقط من إذاعة رزم غير ضرورية، ولكن ولأول مرة يمكن أن تستخدم الشجرة الممتدة من قِبَل أي عقدة لبدء إرسال إذاعي، كما هو موضح في الشكل 4-46. لاحظ أنه ليس من الضروري أن تكون العقدة على دراية كاملة بالشجرة؛ ببساطة تحتاج فقط لمعرفة أيٍّ من جيرانها في G هم جيران على الشجرة الممتدة.

التعقيد الأساسي لطريقة الشجرة الممتدة هو بناء وصيانة الشجرة. تم تطوير العديد من خوارزميات الشجرة الممتدة الموزّعة [Gallager 1983; Gartner 2003]. سنذكر هنا فقط خوارزمية بسيطة. تعتمد هذه الطريقة على وجود عقدة مركزية معرّفة – تُعرف أيضاً بنقطة الالتقاء (rendezvous point) أو نقطة القلب (core

(tree-join point) – لبناء شجرة ممتدة. تُرسل العقد رسائل التحاق بالشجرة (messages معنونة إلى العقدة المركزية. يتم تمرير رسالة الالتحاق باستخدام توجيه unicast نحو المركز حتى تصل إلى عقدة تنتمي للشجرة الممتدة أو تصل إلى المركز. في كلتا الحالتين يحدد المسار الذي اتبعته رسالة الالتحاق الفرع من الشجرة بين العقدة الطرفية التي بدأت الرسالة والمركز. يمكن أن نعتبر هذا المسار الجديد على أنه "مُطَعَّم" (ملحق) للشجرة الممتدة الموجودة.

يوضح الشكل 4-47 بناء شجرة ممتدة مركزية. افترض أن العقدة E اختيرت كمركز للشجرة، وافترض أن العقدة F هي العقدة الأولى التي تنضم للشجرة بإرسال رسالة التحاق إلى E. تصبح الوصلة الوحيدة EF هي الشجرة الممتدة الأولى. ثمّ تنضم العقدة B إلى الشجرة بإرسال رسالة التحاق نحو E. افترض أن المسار من B إلى E هو عن طريق D. في هذه الحالة ستؤدّي رسالة الالتحاق إلى أن يطعّم المسار BDE في الشجرة. بعد ذلك تنضم العقدة A بإرسال رسالة التحاق نحو E. إذا كان المسار من A إلى E يمر بـ B، ولأن B موجودة بالشجرة بالفعل فسيؤدي وصول رسالة A عند B إلى أن يطعّم المسار AB على الفور في الشجرة. تنضمّ العقدة C إلى الشجرة بإرسال رسالة نحو E. أخيراً لأن توجيه المسار من G إلى E يجب أن يكون عن طريق العقدة D، فإنه عندما ترسل G رسالة إلى E، ستضاف الوصلة GD إلى الشجرة.



الشكل 4-47 بناء شجرة ممتدة مركزية.

خوارزميات الإذاعة في الواقع العملي

تُستخدم بروتوكولات الإذاعة عملياً في طبقتي الشبكة والتطبيقات. كما أوردنا في الجزء 2-6 يستخدم بروتوكول Gnutella [Gnutella 2007] الإرسال الإذاعي في طبقة التطبيقات لإذاعة استفسارات المحتوى بين نظائر Gnutella. في هذه الحالة تُمثَّل الوصلة بين عمليتي نظير في مستوى التطبيقات موزّعتين في شبكة Gnutella في الواقع بتوصيلة TCP. يستخدم Gnutella الفيض المحكوم بالرقم المتسلسل حيث يستعمل 16 بتاً للمعرّف و16 بتاً لوصف نوع الحمل الآجر (والذي يعرّف نوع رسالة Gnutella) في اكتشاف ما إذا كان استفسار الإذاعة الذي تم استلامه قد سبق استلامه ونسخه وإرساله. كما ذكرنا أيضاً في الجزء 2-6 يستخدم بروتوكول Gnutella أيضاً حقل زمن فترة العمر (TTL) لتحديد عدد القفزات التي سيرسل عليها الاستفسار. عندما تستلم عملية Gnutella استفساراً وتنسخه فإنها تُنقِص حقل TTL قبل إرسال الاستفسار. وهكذا سيصل استفسار Gnutella فقط إلى النظائر التي تقع ضمن عدد معين (القيمة الأولية لـ TTL) من القفزات في مستوى التطبيقات ابتداءً من مصدر الاستفسار. ولذا يطلق أحياناً على آلية فيض Gnutella اسم فيض المجال المحدود.

يُستخدم أيضاً شكلٌ من الفيض المحكوم بأرقام متسلسلة لإذاعة إعلانات حالة الوصلة (LSAs) ضمن خوارزمية التوجيه في بروتوكول OSPF [RFC 2328; Perlman 1999] وبروتوكول IS-IS [RFC 1142; Perlman 1999]. يستعمل OSPF عدداً مكوناً من 32 بتاً كرقم متسلسل، بالإضافة إلى حقل العمر (age field) المكون من 16 بتاً لتمييز حالة الوصلة LSAs. تذكّر أن عقدة OSPF تذيع LSAs لوصلاتها الملحقة بشكلٍ دوري، أو عندما تتغيّر كلفة وصلة إلى عقدة مجاورة ، أو عندما تتغيّر حالة الوصلة إلى شغّالة أو متعطلة. تُستخدم الأرقام المتسلسلة لرزم LSA لاكتشاف تكرار LSAs، ولكنها تؤدي أيضاً وظيفة مهمة أخرى في بروتوكول OSPF. من المحتمل أثناء استخدام الفيض أن تصل رزمة LSA انشئت بالمصدر في الوقت t بعد رزمة LSA أخرى اُنشئت بنفس المصدر في الوقت

$\delta+t$. تسمح الأرقام المتسلسلة المستخدمة لدى عقدة المصدر بتمييز LSA الأقدم عن LSA الأحدث. يخدم حقل العمر غرضاً مشابهاً لقيمة TTL. تبدأ قيمة حقل العمر الأولية بصفر وتزاد في كل قفزة، كما تزداد أيضاً أثناء وجود الرزمة في ذاكرة الموجّه بانتظار فيضها. ورغم أننا وصفنا خوارزمية فيض LSA فقط سريعاً هنا، فإننا نلاحظ أن تصميم بروتوكولات إذاعة LSA يمكن أن يكون عملاً صعباً جدا في الواقع. يمكنك الاطلاع على حادثة موصوفة في [RFC 789؛ Perlman 1999] أدى فيها إرسال فيض LSAs بشكلٍ خاطئ من موجّهين معطوبين إلى تعطل شبكة ARPAnet بالكامل!

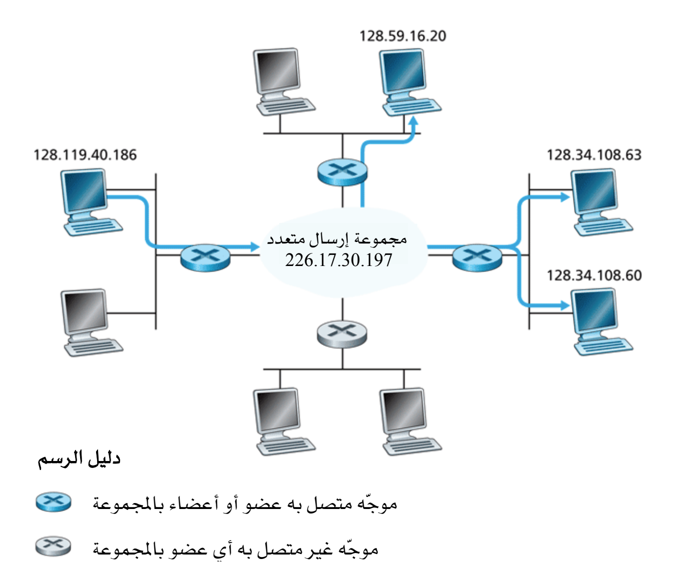
4-7-2 الإرسال المتعدد (الجماعي)

رأينا في الجزء السابق أن خدمة الإرسال الإذاعي توصِّل الرزم إلى كل عقدة في الشبكة. سنحوّل انتباهنا في هذا الجزء إلى خدمة الإرسال المتعدد (الجماعي) (multicast service) والتي يتم فيها توصيل الرزمة إلى مجموعة جزئية فقط من العقد الموجودة بالشبكة. يتطلب عددٌ من التطبيقات الحديثة للشبكة توصيل الرزم من مُرسِل واحد أو أكثر إلى مجموعة من المستلمين. تشمل هذه التطبيقات نقل كتل ضخمة من البيانات (كنقل ترقية للبرامج من مطوّرها إلى المستخدمين الذين يحتاجونها)، وكذلك العرض المستمر لمواد الوسائط المتعددة (كنقل التسجيل الصوتي والفيديو والنصوص لمحاضرة حية إلى مجموعة مشاركين موزعين في أماكن متفرقة)، وتطبيقات المشاركة في البيانات (مثل السبورة (whiteboard) والمؤتمرات عن بُعد (teleconferencing) بين مشاركين كُثُر موزعين في أماكن متفرقة)، وكذلك مصادر تغذية البيانات (data feeds) (كأسعار الأسهم)، وتحديث ذاكرة الويب الوسيطة والمحادثات التفاعلية (كالبيئات الافتراضية التفاعلية الموزّعة أوالألعاب متعددة اللاعبين).

في الاتصالات الجماعية تواجهنا مباشرة مشكلتان: كيف يمكن تمييز المستلمين لرزمة جماعية، وكيف تُعنون رزمة مرسلة إلى هؤلاء المستلمين. في حالة الاتصال الفردي (unicast) يوضع عنوان IP مميز للمستلم (الوجهة) في كل رزمة

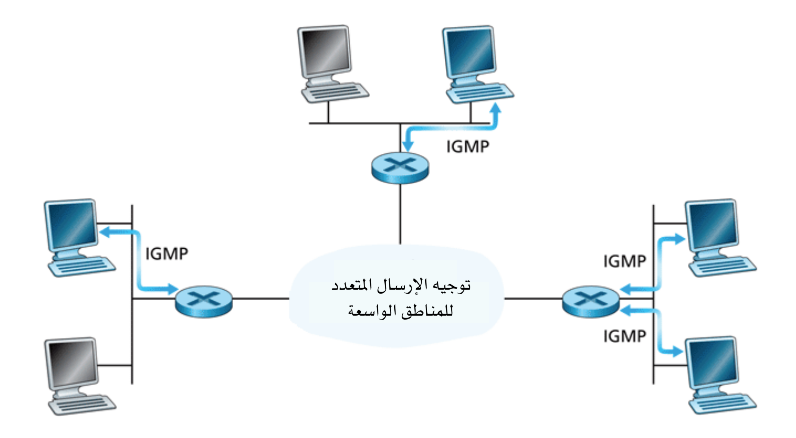
بيانات. أما في حالة الإرسال الإذاعي فتحتاج كل العقد لاستلام الرزمة المذاعة، ولذا لا نحتاج إلى عناوين للوجهة. لكن في حالة الإرسال الجماعي عندنا الآن مستلمون متعدّدون. هل يعقل أن تحمل كل رزمة جماعية عناوين IP لجميع المستلمين؟ ربما تكون هذه الطريقة عملية مع عدد صغير من المستلمين، لكنها غير قابلة للتوسع بطريقة جيدة للتعامل مع مئات أو آلاف المستلمين، حيث إن كمية معلومات العناوين في رزمة البيانات ستفوق بكثير كمية البيانات الموجودة حقيقةً في حقل الحمل الآجر للرزمة. يتطلب التعريف الصريح للمستلمين من قِبَل المُرسِل أيضاً أن يعرف المُرسِل هويّات وعناوين كل المستلمين. سنرى بعد قليل أن هناك حالات يكون فيها هذا المطلب غير مرغوب.

لهذه الأسباب تُعنوَن رزم الإرسال الجماعي في بنية الإنترنت (وبنى شبكات أخرى كمكائن سحب النقود [Black 1995]) باستخدام العنونة غير المباشرة. بمعنى أنه يكفي استعمال معرّف واحد لمجموعة المستلمين، وترسل نسخة واحدة من الرزمة معنونة إلى المجموعة باستعمال هذا المعرّف الوحيد إلى كل المستلمين المشتركين بتلك المجموعة. في الإنترنت يمثل المعرّف الوحيد لمجموعة من المستلمين بعنوان IP من الفئة D. يطلق على مجموعة المستلمين المرتبطة بعنوان IP من الفئة D مجموعة الإرسال الجماعي (multicast group). يوضح الشكل 4-48 المفهوم التجريدي لمجموعة الإرسال الجماعي، حيث ترتبط أربعة مضيفات (موضحة بلون مظلّل) بالعنوان الجماعي 226.17.30.197 وسوف تستلم كل وحدات البيانات المعنونة إلى تلك المجموعة. المشكلة التي لا يزال علينا تناولها هي حقيقة أن كل مضيف له عنوان IP فردي فريد ومستقل جداً عن عنوان المجموعة التي يشارك فيها.



الشكل 4-48 مجموعة الإرسال الجماعي: رزمة بيانات معنونة إلى المجموعة يتم تسليمها إلى كل أعضاء المجموعة.

بينما يبدو تجريد مجموعة الإرسال الجماعي أمراً بسيطاً فإنه يثير في الواقع العديد من التساؤلات. كيف تبدأ مجموعة وكيف تنتهي؟ كيف يُختار عنوان المجموعة؟ كيف تلتحق مضيفات جديدة بالمجموعة (كمُرسِلين أو كمستقبلين)؟ هل بالإمكان أن ينضم أي واحد إلى مجموعة ما (ويرسل لها أو يستقبل منها) أم أن عضوية المجموعة مقيّدة، وإذا كان الأمر كذلك فمن قِبَل مَنْ يتم ذلك التقييد؟ هل يعرف أعضاء المجموعة هويّات أعضاء المجموعة الآخرين كجزء من بروتوكول طبقة الشبكة؟ كيف تعمل عقد الشبكة مع بعضها البعض لتسليم رزمة بيانات إلى كل أعضاء المجموعة؟ في شبكة الإنترنت تكمن الإجابات على كل هذه الأسئلة في بروتوكول إدارة المجموعات للإنترنت (IGMP) (Internet Group Management Protocol) [RFC 3376]. لذا سنعرج سريعاً على بروتوكول IGMP ثم نعود بعد ذلك إلى تلك الأسئلة العامة.



الشكل 4-49 مكونا طبقة الشبكة للإرسال الجماعي في الإنترنت: بروتوكول IGMP وبروتوكولات التوجيه للإرسال الجماعي.

بروتوكول إدارة مجموعة للإنترنت (IGMP)

تعمل النسخة الثالثة من بروتوكول IGMP بين مضيف وموجّه ملحق به مباشرةً (وبشكلٍ غير رسمي يمكن أن نفكّر بالموجّه الملحق مباشرةً كموجه أول قفزة (first hop router) الذي يراه المضيف على المسار إلى أي مضيف آخر خارج شبكته المحلية الخاصة، أو موجّه القفزة الأخيرة على أي مسار إلى ذلك المضيف). يوضح الشكل 4-49 ثلاثة موجّهات قفزة أولى للإرسال الجماعي، كلٌّ منها مرتبط مع مضيفاته الملحقة عن طريق وصلة محلية خارجة واحدة. ترتبط هذه الوصلة المحلية بشبكة اتصالات محلية في هذا المثال. ورغم وجود العديد من المضيفات بكل شبكة اتصالات محلية إلا أنه غالباً ما ينتمي فقط عدد قليل من تلك المضيفات على أقصى تقدير إلى مجموعة معينة للإرسال الجماعي في أي لحظة.

يوفر بروتوكول IGMP وسائل للمضيف لإعلام الموجّه الملحق بأن تطبيقاً ما يجري تنفيذه على المضيف يريد الانضمام إلى مجموعة معينة للإرسال الجماعي (multicast group). ونظراً لأن مجال تفاعل IGMP محدود بين المضيف والموجّه الملحق به، فمن الواضح أن هناك حاجة لبروتوكول آخر للتنسيق بين موجّهات الإرسال الجماعي (بما في ذلك الموجّهات الملحقة) في كافة أنحاء الإنترنت لكي

توجه وحدات بيانات الإرسال الجماعي إلى وجهاتها النهائية. تتحقق هذه الوظيفة الأخيرة عن طريق خوارزميات طبقة الشبكة للتوجيه الجماعي (كتلك التي سنناقشها بعد قليل). وهكذا يتضمن الإرسال الجماعي في طبقة الشبكة في الإنترنت مكوّنين متكاملين: بروتوكول IGMP وبروتوكولات التوجيه للإرسال الجماعي.

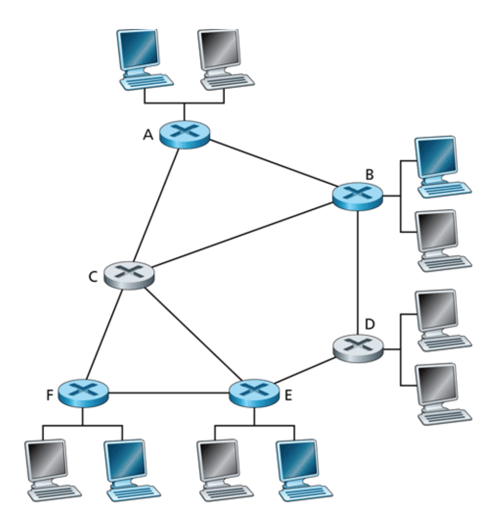
يتضمن بروتوكول IGMP ثلاثة أنواع فقط من الرسائل. كما هو الحال مع بروتوكول ICMP، تغلف رسائل IGMP ضمن وحدات بيانات IP وتحمل القيمة 2 في حقل رقم البروتوكول. ترسل رسالة استفسار العضوية (membership_query) من قِبَل موجّه إلى كل المضيفات على واجهة ملحقة (مثلاً إلى كل المضيفات على شبكة محلية) لتحديد المجموعات المختلفة التي انضمت إليها مضيفات على تلك الواجهة. ترد المضيفات على رسالة استفسار العضوية برسالة تقرير (membership_report) عن أعضاء IGMP. يمكن أيضاً إنشاء رسائل تقرير العضوية من قِبَل مضيف عند بداية انضمام تطبيق إلى مجموعة بدون انتظار لرسالة استفسار العضوية من الموجّه. النوع الثالث والأخير من رسائل IGMP هو رسالة مغادرة المجموعة (leave_group)، وهي رسالة اختيارية. لكن إذا كانت اختيارية فكيف يكتشف موجّه أن مضيفاً قد غادر مجموعة الاتصال الجماعي؟ الجواب على ذلك هو أن الموجّه يستنتج أن المضيف لم يعد في المجموعة إذا لم يرد على رسالة استفسار بالعنوان المحدد للمجموعة. هذا مثال لما يسمى أحياناً بـ "الحالة المرنة" (soft state) في بروتوكولات الإنترنت. في بروتوكول ذي "حالة مرنة" تُحذَف الحالة (في IGMP تشير الحالة إلى حقيقة أن هناك مضيفات ملحقة بمجموعة اتصالٍ جماعي معينة) عن طريق حدث "انتهاء الوقت" (timeout) (في هذه الحالة عن طريق رسائل دورية من الموجّه للاستفسار عن العضوية) إذا لم يتم تحديثها بشكلٍ واضح (في هذه الحالة عن طريق رسائل تقرير العضوية من مضيف ملحق). يرجع السبب في استخدام بروتوكولات الحالة المرنة إلى أنها تؤدي إلى تحكم أسهل من بروتوكولات الحالة المحددة (hard-state) والتي تتطلّب ليس فقط إضافة وإزالة الحالة بشكلٍ محدد ولكن أيضاً آليات للتعافي من الوضع الذي ينسحب فيه الكيان المسؤول عن إزالة

الحالة قبل الأوان أو يطرأ عليه عطب. يمكنك الاطلاع على مناقشات مفيدة عن الحالة المرنة في [Lui 2004; Raman 1999; Ji 2003].

خوارزميات التوجيه للإرسال الجماعي

يوضح الشكل 4-50 مشكلة التوجيه للإرسال الجماعي، وفيه تظهر المضيفات الملتحقة بمجموعة، وكذلك الموجّه الذي تلتحق عن طريقه مباشرة مظللة اللون. كما هو موضح في الشكل 4-50 تحتاج في الواقع مجموعة جزئية فقط من الموجّهات (تلك التي لديها مضيفات ملتحقة بمجموعة الإرسال الجماعي) لاستلام وحدات بيانات الإرسال الجماعي. في الشكل 4-50 تحتاج الموجّهات A، B، E، F فقط لاستلام وحدات بيانات الإرسال الجماعي. ولأنه لا يوجد أيٌّ من المضيفات المتصلة بالموجّه D ملتحق بمجموعة الإرسال الجماعي، ولأن الموجّه C لا يرتبط به أي مضيفات أصلاً، لذا لا يحتاج C ولا D لاستلام حركة مرور الإرسال الجماعي. إن هدف التوجيه للإرسال الجماعي إيجاد شجرة الوصلات التي تربط بين كل الموجّهات التي لديها مضيفات منضمة إلى مجموعة الإرسال الجماعي. بعد ذلك توجّه رزم الإرسال الجماعي خلال تلك الشجرة من المُرسِل إلى كل المضيفات المرتبطة بالشجرة. بالطبع قد تتضمن الشجرة موجّهات ليس لديها مضيفات ملتحقة بمجموعة الإرسال الجماعي (على سبيل المثال في الشكل 4-50 من المستحيل توصيل الموجّهات A، B، E، F في شجرة بدون المرور على الموجّه C أو الموجّه D).

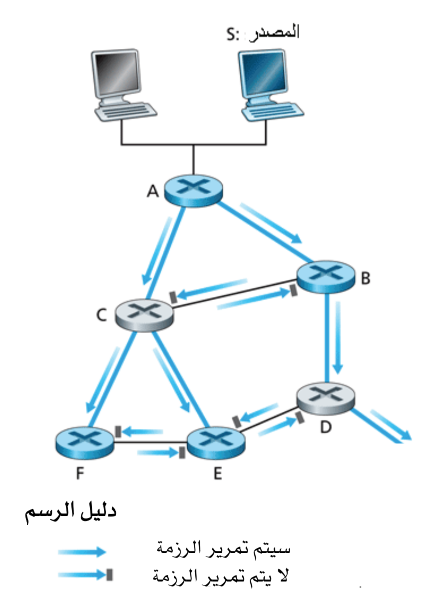
عملياً تم تبني طريقتين لتحديد شجرة توجيه الإرسال الجماعي، ولقد درسنا كلتيهما ضمن سياق توجيه البث الإذاعي، ولذا سنكتفي بذكرهما فقط هنا. تختلف الطريقتان طبقاً لنوع الشجرة المستخدمة لتوزيع وحدات بيانات الإرسال الجماعي: هل هي شجرة وحيدة مشتركة لكل المجموعة بغض النظر عن عدد المصادر (group-shared tree) أو أن كل مصدر يبني شجرة توجيه خاصة به للإرسال إلى المجموعة (source-specific routing tree).



الشكل 4-50 مضيفات الإرسال الجماعي والموجّهات الملحقة بها والموجّهات الأخرى.

- توجيه الإرسال الجماعي باستخدام شجرة مشتركة للمجموعة: كما في حالة الإذاعة عبر الشجرة الممتدة (spanning-tree broadcast)، يعتمد توجيه الإرسال الجماعي باستخدام شجرة مشتركة للمجموعة على بناء شجرة تتضمّن كل موجّهات الأطراف التي لديها مضيفات ملتحقة بمجموعة الإرسال الجماعي. عملياً تستخدم طريقة مركزية لبناء شجرة التوجيه وذلك بجعل موجّهات الأطراف التي لديها مضيفات ملتحقة تُرسِل (عن طريق الإرسال الفردي unicast) رسائل التحاق معنونة إلى عقدة المركز. كما في حالة الإذاعة توجّه رسالة الالتحاق عن طريق توجيه الإرسال الفردي نحو المركز حتى تصل الرسالة إلى موجّه ينتمي حالياً إلى الشجرة أو تصل إلى المركز. كل الموجّهات على طول الطريق الذي تعبر عليه رسالة الالتحاق سوف توجّه وحدات بيانات الإرسال الجماعي إلى موجهات الأطراف تلك التي بدأت رسالة الالتحاق. وهناك سؤال مهم لتوجيه الإرسال الجماعي المعتمد على شجرة مركزية ألا وهو: كيف يمكن اختيار المركز؟ توجد مناقشات حول خوارزميات اختيار المركز في [Wall 1980; Thaler 1997; Estrin 1997].
- توجيه الإرسال الجماعي باستخدام شجرة لكل مصدر: بينما يبنى توجيه الإرسال الجماعي باستخدام شجرة مشتركة للمجموعة (أي شجرة واحدة)

لتوجيه الرزم من كل المصادر، تستخدم الطريقة الثانية شجرة توجيه لكل مصدر في مجموعة الإرسال الجماعي. عملياً تستخدم خوارزمية RPF (بعقدة المصدر x) لبناء شجرة لتوجيه رزم البيانات المتولدة في المصدر x. تتطلب خوارزمية RPF للإذاعة التي درسناها سابقاً تعديلاً بسيطاً لاستخدامها في الإرسال الجماعي. ولتفهم سبب ذلك خذ في الاعتبار الموجّه D في الشكل 4-51. في حالة بروتوكول RPF للإذاعة يرسل الموجّه D الرزم إلى الموجّه G بالرغم من أن الموجّه G ليس لديه مضيفات ملحقة ضمن المجموعة. بينما هذا ليس أمراً سيئاً جداً في هذه الحالة حيث إن D عنده موجّه واحد فقط في اتجاه الإرسال (G)، تخيل ما يحدث إذا كان هناك آلاف الموجّهات في اتجاه الإرسال من D؛ ستستلم كل هذه الآلاف من الموجّهات رزماً غير مرغوبة. (هذا السيناريو ليس متكلّفاً كما قد يبدو. فقد عانت أول شبكة عالمية للإرسال الجماعي والمعروفة بـ MBone [Casner 1992; Macedonia 1994] في البداية من هذه المشكلة بالضبط). الحلّ لمشكلة استلام رزم غير مرغوبة عند استخدام RPF هو استخدام ما يُعرف بالتقليم (pruning). عندما يستلم موجّه رزم الإرسال الجماعي وليس لديه مضيفات ملحقة منضمّة لتلك المجموعة فإنه يرسل رسالة prune إلى الموجّه الأعلى (عكس اتجاه الإرسال). إذا استلم موجّه رسالة prune من كلٍّ من موجّهاته في اتجاه الإرسال فعندئذ يمكنه تمرير رسالة prune للموجّه الأعلى منه في عكس اتجاه الإرسال.



الشكل 4-51 تمرير المسار العكسي في حالة الإرسال الجماعي.

توجيه الإرسال الجماعي في الإنترنت

كان بروتوكول متجه المسافة لتوجيه الإرسال الجماعي (DVMRP) أول بروتوكول لتوجيه الإرسال الجماعي في الإنترنت [RFC 1075]. يستخدم بروتوكول DVMRP شجرةً لكل مصدر، كما يستخدم خوارزمية RPF مع التقليم (pruning) والتي ناقشناها من قبل. ربما يكون البروتوكول الأكثر استخداماً على نطاق واسع في الإنترنت لتوجيه الإرسال الجماعي هو بروتوكول PIM (Protocol Independent Multicast)، والذي يُعرِّف بشكلٍ صريح سيناريوهين لتوزيع وحدات البيانات. يعرف الأول بالنمط الكثيف (dense mode) [RFC 3973]، وفيه توجد مواقع أعضاء المجموعة بشكلٍ مكثف، وهذا يعني بالضرورة أن الكثير من الموجّهات الموجودة في منطقة أو أغلبها تشترك في توجيه وحدات البيانات. ويُعدّ نمط PIM الكثيف أسلوب تمرير مسار عكسي يستخدم الفيض والتقليم ويشبه من حيث المبدأ بروتوكول DVMRP.

يُعرَف النمط الثاني بالنمط المتناثر (sparse mode) [RFC 4601]، حيث يكون عدد الموجّهات المرتبطة بأعضاءٍ في المجموعة قليلاً مقارنةً بالعدد الكلي للموجّهات، ويكون أعضاء المجموعة متفرّقين على نحو واسع. يُستخدم بروتوكول PIM ذو النمط المتناثر نقاط الالتقاء (rendezvous points) لإعداد شجرة توزيع الإرسال الجماعي. أما في الإرسال الجماعي المحدد بمصدر معين ((Source-Specific (SSM) (Multicast [RFC 3569; RFC 4607] فيُسمَح لمُرسِل واحد فقط بالإرسال خلال الشجرة مما يبسّط إلى حدٍّ كبير إنشاء وصيانة الشجرة.

عند استخدام بروتوكولي PIM وDVMP ضمن نطاق معين يمكن أن يهيئ مشغل الشبكة موجّهات IP الموجودة داخل ذلك النطاق للإرسال الجماعي بنفس الطريقة تقريباً التي يهيئ بها بروتوكولات التوجيه الفردي داخل النطاق مثل RIP وIS-IS وOSPF. لكن ماذا يحدث عندما تكون مسارات الإرسال الجماعي مطلوبة بين النطاقات المختلفة؟ هل هناك بروتوكول إرسال جماعى مكافئ لبروتوكول BGP بين النطاقات؟ إن الجواب (بشكلٍ حرفي) نعم. في [RFC 4271] تم تعريف امتدادات متعددة لبروتوكول BGP للسماح له بنقل معلومات التوجيه للبروتوكولات الأخرى بما في ذلك معلومات الإرسال الجماعي. مثلاً يمكن استخدام بروتوكول اكتشاف مصدر الإرسال الجماعي (Multicast Source Discovery Protocol (MSDP)) [RFC 3618; RFC 4611] لتوصيل نقاط التقاء ببعضها البعض في نطاقات مختلفة من بروتوكول نمط PIM المتناثر.

دعنا ننهي مناقشتنا للإرسال الجماعي في الإنترنت بملاحظة أنه لا يزال أمامنا الكثير لتحقيق الاستفادة من الإرسال الجماعي في الإنترنت على نحوٍ كبير. يمكنك الاطلاع على المناقشات الممتعة حول النموذج الحالي لخدمة الإرسال الجماعي في الإنترنت وما يتعلق بقضايا انتشارها في [Diot 2000; Sharma 2003]. ومع ذلك ورغم عدم الانتشار الواسع إلا أن الإرسال الجماعى على مستوى الشبكة أبعد ما يكون من وصفه بأنه "عديم الفائدة". فحركة مرور بيانات الإرسال الجماعي موجودة على Internet 2 والشبكات المرتبطة بها منذ عدة سنوات [Internet2 Multicast 2007]. قامت إذاعة BBC في المملكة المتّحدة بتوزيع المحتوى

عن طريق الإرسال الجماعي على الإنترنت [BBC Multicast 2007]. في نفس الوقت وكما رأينا مع PPLive في الفصل الثاني وفي أنظمة النظائر الأخرى كالإرسال الجماعي بين الأنظمة الطرفية (End System Multicast) [ESM 2007] يوفر الإرسال الجماعي في مستوى طبقة التطبيقات خدمة الإرسال الجماعي للمحتوى بين النظائر باستخدام بروتوكولات الإرسال الجماعي في طبقة التطبيقات (بدلاً من طبقة الشبكة). هل ستطبّق خدمات الإرسال الجماعي في المُستقبل بشكلٍ أساسي في طبقة الشبكة (في قلب الشبكة) أم في طبقة التطبيقات (على أطراف الشبكة)؟ رغم أن الهوس الحالي لتوزيع المحتوى عن طريق أساليب النظائر يرجّح – على الأقل في المستقبل القريب – الكفّة تجاه تطبيق الإرسال الجماعي في طبقة التطبيقات، إلا أن التقدّم مازال مستمراً في تطبيق الإرسال الجماعي في طبقة الشبكة للإنترنت - وأحياناً يحرز قصب السبق في النهاية البطيئ الذي يسير بخطى ثابتة.

8-4 الخلاصة

بدأنا رحلتنا في هذا الفصل إلى صميم قلب الشبكة (network core) وعرفنا أن طبقة الشبكة توجد على كل مضيف وموجّه في الشبكة، ولذا تُعدّ بروتوكولات طبقة الشبكة من بين البروتوكولات الأكثر صعوبةً في رصّة البروتوكولات.

كما عرفنا أن الموجّه قد يحتاج لمعالجة الملايين من تدفقات الرزم بين أزواج مختلفة من المصدر والوجهة في نفس الوقت. للسماح لموجّه بمعالجة مثل هذا العدد الكبير من التدفقات، تعلم مصممو الشبكة على مر السنين أن مهام الموجّه يجب أن تكون بسيطة بقدر الإمكان. يمكن أن تؤخذ العديد من الإجراءات لتسهيل عمل الموجّه كاستخدام طبقة شبكة تتعامل مع وحدات البيانات بدلاً من طبقة شبكة ذات دوائر افتراضية، واستعمال ترويسة انسيابية وذات حجم ثابت (كما في IPv6)، وعدم السماح بتجزئة وحدة البيانات (وهذا أيضاً متاح في بروتوكول IPv6)، وتوفير خدمة وحيدة وفريدة تتمثل في "خدمة أفضل جهد". لعل الخدعة الأكثر أهمية هنا هي عدم التتبع الفردي لكل تدفق على حدة، وإنما جعل قرارات

التوجيه تعتمد فقط على عناوين وجهات منظمة بشكلٍ هرمي في وحدات البيانات. من المفيد ملاحظة أن الخدمة البريدية تستخدم هذه الطريقة منذ عدّة سنوات.

استعرضنا في هذا الفصل أيضاً المبادئ التي تُبنى عليها خوارزميات التوجيه، وتعلمنا كيف تجرّد خوارزميات التوجيه شبكة الحاسب إلى رسم بياني (graph) مكون من عقد ووصلات. بهذا التجريد يمكننا توظيف النظرية الغنية لأقصر مسار (shortest path) للتوجيه خلال الرسوم البيانية، والتي طوّرت خلال السنوات الأربعين الماضية في مجتمعات الخوارزميات وبحوث العمليات. رأينا أن هناك طريقتين رئيستين: الطريقة المركزية (العالمية) — وفيها تحصل كل عقدة على خريطة كاملة للشبكة وتطبق بشكلٍ مستقل خوارزمية توجيه المسار الأقصر؛ والطريقة اللامركزية — وفيها تحصل كل عقدة بشكلٍ فردي على صورة جزئية للشبكة، ورغم ذلك تعمل العقد سويةً لتوصيل الرزم على أقصر المسارات. درسنا أيضا كيف تُستخدم التراكيب الهرمية للتعامل مع مشكلة التوسع وذلك بتقسيم الشبكات الكبيرة إلى مناطق إدارية مستقلة تسمى النظم المستقلة ذاتياً (ASs). يقوم كل نظام AS بتوجيه وحدات البيانات داخله بشكلٍ مستقل تماماً كما تفعل كل دولة لتوجيه رسائل البريد داخلها بشكلٍ مستقل. وتعلمنا كيف تم تجسيد الأساليب المركزية واللامركزية والتركيب الهرمي في بروتوكولات التوجيه الرئيسة المستخدمة في الإنترنت مثل RIP وOSPF وBGP. وختمنا دراسة خوارزميات التوجيه بمناقشة توجيه البث الإذاعي (العام) والتوجيه المتعدد (الجماعي).

بهذا تكتمل دراستنا لطبقة الشبكة وتستمر رحلتنا في الفصل القادم بالنزول خطوة واحدة خلال رصّة البروتوكولات لمناقشة قضايا طبقة ربط البيانات. وكما هو الحال في طبقة الشبكة تُعدّ طبقة ربط البيانات أيضاً جزءاً من صميم قلب الشبكة. غير أننا سنرى أن طبقة ربط البيانات لها مهمة أكثر محلية لنقل الرزم بين العقد على نفس الوصلة أو شبكة الاتصالات المحلية. وبالرغم من أن هذه المهمة قد تظهر على السطح بأنها بسيطة بالمقارنة بمهام طبقة الشبكة، إلا أننا سنرى أن طبقة ربط البيانات تتضمّن عدداً من القضايا المهمة والممتعة والتي يمكن أن تستغرق منا وقتاً طويلاً لدراستها.

أسئلة وتمارين وتدريبات الفصل الرابع

❖ أسـئلة مراجعة

• الأجزاء 4-1 حتى 4-2

1. دعنا نراجع بعض المصطلحات المستخدمة في هذا الكتاب. تذكر أن رزمة طبقة النقل يطلق عليها "قطعة" (segment)، ورزمة طبقة ربط البيانات يطلق عليها "إطار" (frame). فماذا يطلق على رزمة طبقة الشبكة؟ وتذكر أنه يطلق على كلٍّ من الموجّهات ومحوّلات طبقة ربط البيانات " محوّلات الرزم" (packet switches)؛ فما الفرق الأساسي بين الموجّه ومحوّل طبقة ربط البيانات؟ وتذكر أيضاً أن نستخدم المصطلح "موجّه" لكلٍّ من شبكات وحدات البيانات وشبكات الدوائر الافتراضية.
2. ما الوظيفتان الأكثر أهمية لطبقة الشبكة في شبكات وحدات البيانات؟ وما تلك في شبكات الدوائر الافتراضية؟
3. ما الفرق بين التوجيه (routing) والتمرير (forwarding)؟
4. هل تستخدم الموجّهات في كلٍّ من شبكات وحدات البيانات وشبكات الدوائر الافتراضية جداول تمرير؟ وإذا كان الأمر كذلك فصف تلك الجداول في كلتا الحالتين.
5. صف بعض الخدمات المحتملة التي يمكن أن توفرها طبقة الشبكة لرزمة معينة، وكذلك التي توفرها لتدفق من الرزم (packet flow). هل أيٌّ من تلك الخدمات المحتملة متوفر في طبقة الشبكة للإنترنت؟ وهل أيٌّ منها متوفر في نموذج خدمة معدل البتات الثابت (CBR) لشبكة ATM؟
6. اذكر بعض التطبيقات التي يمكن أن تستفيد من نموذج خدمة معدل البتات الثابت (CBR) لشبكة ATM.

• الجزء 4-3

7. ناقش أسباب تخزين كل منفذ من منافذ المدخل (input ports) في الموجّه عالي السرعة نسخة ظلّ (shadow copy) من جدول التمرير.
8. تم مناقشة ثلاثة أنواع من أنسجة المحوّلات في الجزء 4-3؛ اذكر كل نوع مع وصفه باختصار.

9. صف كيف يمكن أن يحدث الفقد في الرزم عند منافذ المدخل، وكيف يمكن التخلص من ذلك (بدون استخدام مخازن مؤقتة بسعة لا نهائية).
10. صف كيف يمكن أن يحدث الفقد في الرزم عند منافذ المخرج.
11. ما المقصود بحجب HOL؟ وهل يحدث عند منافذ المدخل أم عند منافذ المخرج؟

- **الجزء 4-4**

12. هل يخصص للموجّهات عناوين IP؟ وإذا كان الأمر كذلك، فكم عددها؟
13. ما العدد الثنائي المؤلّف من 32 بتاً والمكافئ لعنوان IP التالي: 223.1.3.27؟
14. قم بزيارة أحد المضيفات الذي يستخدم بروتوكول DHCP للحصول على عنوان IP وقناع الشبكة والموجّه الافتراضي وعنوان خادم DNS المحلي، ومن ثم اكتب تلك القيم.
15. افترض وجود ثلاث موجّهات بين مضيف مصدر ومضيف الوجهة له. بإهمال تجزئة الرزمة (fragmentation) فكم عدد الواجهات (interfaces) التي تمر خلالها وحدة البيانات من المصدر إلى الوجهة؟ وكم عدد جداول التمرير التي سيتم البحث فيها لنقل وحدة البيانات من المصدر إلى الوجهة؟
16. افترض أن تطبيقاً يولّد قطع بيانات مؤلفة من 40 بايتاً كل 20 ميلي ثانية، وأن كل قطعة بيانات يتم تغليفها في قطعة TCP ثم في وحدة بيانات IP. ما النسبة المئوية من كل وحدة بيانات تُعتبر عبئاً إضافياً؟
17. افترض أن المضيف A يرسل إلى المضيف B قطعة TCP مُغلّفة في وحدة بيانات IP. عندما يستلم المضيف B وحدة البيانات فكيف تعرف طبقة الشبكة في المضيف B أنه يجب أن تُسلَم القطعة (أي الحمل الآجر لوحدة بيانات IP) لبروتوكول TCP وليس UDP أو أي شيء آخر؟
18. افترض أنك قمت بشراء موجّه لاسلكي وقمت بتوصيله بمودم الكبل لديك. وافترض أيضاً أن موفر خدمة الإنترنت لك يخصص بشكلٍ ديناميكي لجهازك الموصل (أي موجّه اللاسلكي) عنوان IP واحد. أيضاً افترض أن لديك خمسة حاسبات شخصية تستخدم بروتوكول 802.11 للاتصال بالموجّه لاسلكياً. كيف يتم تخصيص عناوين IP لتلك الحاسبات الخمسة؟ هل يستخدم الموجّه اللاسلكي NAT؟ بيّن السبب.
19. قارن بين حقول الترويسة لبروتوكول IPv4 وIPv6؛ هل يوجد حقول مشتركة بينهما؟

20.قيل أن IPv6 يعمل أنفاق خلال موجّهات IPv4. هل توافق على أن IPv6 يتعامل مع أنفاق IPv4 تماماً كبروتوكولات طبقة الوصلة (طبقة ربط البيانات)؟ بين سبب الموافقة أو الرفض.

- **الجزء 4-5**

21.قارن بين خوارزمية التوجيه بحالة الوصلة وخوارزمية التوجيه بمتجه المسافة.
22.ناقش كيف سهّل التركيب الهرمي للإنترنت من إمكانية التوسع لتضم الملايين من المُستخدِمين.
23.هل من الضروري أن يستخدم كل نظام مستقل ذاتياً (AS) نفس بروتوكول التوجيه بداخله؟ ما سبب ذلك؟

- **الجزء 4-6**

24.بالنظر إلى الشكل 4-35 وبدءاً من جدول التمرير الأصلي للموجّه D وبافتراض أن D تلقّى من A الإعلان التالي:

الشبكة الفرعية للوجهة	الموجّه التالي	عدد القفزات إلى الوجهة
Z	C	10
W	__	1
X	__	1
....	....	....

فهل سيتغيّر جدول D؟ وإذا كان كذلك فكيف؟

25.قارن بين الإعلانات المستخدمة في كلٍّ من RIP وOSPF.
26.أكمل الجملة: عادةً يقوم RIP بإعلان عدد القفزات للوجهات المختلفة. من ناحية أخرى تقوم تحديثات BGP بالإعلان عن ________________ للوجهات المختلفة.
27.لماذا تستخدم بروتوكولات مختلفة في الإنترنت للتوجيه داخل النظم المستقلة ذاتياً وللتوجيه فيما بينها؟
28.ما سبب أهمية اعتبارات السياسة لبروتوكولات التوجيه داخل النظم المستقلة ذاتياً (مثل OSPF وRIP) كما في بروتوكولات التوجيه فيما بينها (مثل BGP)؟
29. عرّف كلاً من المصطلحات التالية وبيّن العلاقة بينها: شبكة فرعية (subnet)، بادئة (prefix)، مسار BGP.
30. كيف يستخدم BGP خاصيّة NEXT-HOP؟ وكيف يستخدم خاصيّة AS-PATH؟

31. صف كيف يمكن أن يطبق مشرف شبكة موفر خدمة الإنترنت من الطبقة العليا سياسة معينة عند تهيئة بروتوكول BGP؟

- ## الجزء 4-7

32. ما الفرق الأساسي بين تحقيق البث الإذاعي (broadcast) عن طريق عدة إرسالات فردية وبين تحقيقها عن طريق شبكة ذات دعم للبث الإذاعي؟
33. بيّن إذا ما كانت كل جملة من الجمل التالية صحيحة أم خطأ لكلٍ من الطرق الثلاث العامة التي درسناها للبث الإذاعي (الفيض غير المحكوم، الفيض المحكوم، الشجرة الممتدة للإذاعة) (يمكن افتراض عدم فقد أيٍ من الرزم نتيجة فيض المخازن المؤقتة وأن كل الرزم يتم تسليمها على الوصلة بنفس ترتيب إرسالها):
    a. يمكن أن تستلم عقدة عدة نسخ من نفس الرزمة.
    b. يمكن أن تعيد عقدة توجيه عدة نسخ من الرزمة على نفس وصلة المخرج.
34. عندما يلتحق مضيف بمجموعة للإرسال الجماعي فهل يجب أن تغيِّر عنوان IP لها إلى عنوان المجموعة التي تلتحق بها؟
35. ما الأدوار التي يلعبها كلٌّ من بروتوكول IGMP وبروتوكول التوجيه الإرسال الجماعي للمناطق الواسعة؟
36. ما الفرق بين الشجرة المشتركة للمجموعة (group-shared tree) والشجرة لكل مصدر (source-based tree) في سياق توجيه الإرسال الجماعي؟

❖ تمارين

1. لندرس بعض مميزات وعيوب شبكات وحدات البيانات وشبكات الدوائر الافتراضية:
    a. افترض أنه في طبقة الشبكة كانت الموجّهات معرضة لظروف مجهدة والتي قد تُسبب تعطلها بشكلٍ متكرر. بشكلٍ عام ما الذي يجب فعله في هذه الحالة؟ هل هذا يعني أن شبكات وحدات البيانات أفضل من شبكات الدوائر الافتراضية أم العكس في مثل تلك الظروف ؟
    b. افترض أنه لكي توفر ضماناً لمستوى الأداء (مثلاً زمن التأخير) الذي سيُرى خلال المسار من المصدر للوجهة تتطلب الشبكة من المُرسِل أن يحدد المعدل الأقصى لحركة بياناته. إذا كان المعدل المعلن والمعدلات الموجودة حالياً بحيث لا يمكن توصيل البيانات من المصدر إلى الوجهة بشكلٍ يحقق متطلبات التأخير، فإن

المصدر لا يسمح له بالوصول للشبكة. هل مثل هذا الأسلوب سهل التحقيق في بنية شبكات وحدات البيانات أم بنية شبكات الدوائر الافتراضية؟

2. افترض شبكة دائرة افتراضية وأن رقم الدائرة الافتراضية يتألف من 16 بتاً:

a. ما أقصى عدد للدوائر الافتراضية التي يمكن حملها على وصلة ما؟

b. افترض وجود عقدة مركزية تحدد أرقام المسارات والدوائر الافتراضية عند إنشاء التوصيلة. وافترض أن نفس رقم الدائرة الافتراضية يُستَخدم على مسار تلك الدائرة الافتراضية. صف كيف يمكن أن تحدد العقدة المركزية رقم الدائرة الافتراضية عند إنشاء التوصيلة. هل من الممكن عدم وجود رقم دائرة افتراضية مشترك عندما يكون عدد الدوائر الافتراضية الحالية أقل من العدد الأقصى المحدد في الجزء (a)؟

c. افترض أنه على مسار الدائرة الافتراضية يُسمح بأرقام مختلفة على كل وصلة على المسار. أثناء إنشاء توصيلة وبعد تحديد مسار من طرف لطرف، صف كيف تختار الوصلات المختلفة أرقام الدوائر الافتراضية خلالها وكيف تهيئ جداول التمرير الخاصة بها بشكلٍ غير مركزي (أي بدون الاعتماد على عقدة مركزية).

3. يتكون جدول التمرير في شبكة الدائرة الافتراضية أساساً من أربعة أعمدة؛ ما معنى القيم في كلٍّ من هذه الأعمدة؟ ويتكون جدول التمرير في شبكة وحدات البيانات أساساً من عمودين؛ فما معنى القيم في كلٍّ من هذين العمودين؟

4. خذ في الاعتبار شبكة دائرة افتراضية يتكون حقل الدائرة الافتراضية لها من بتين. افترض أن الشبكة تريد أن تُنشئ دائرة افتراضية خلال أربع وصلات: A، B، C، D. افترض أن كلاً من هذه الوصلات تحمل حالياً دائرتين افتراضيتين وأرقامها كما هو مبين بالجدول التالي:

الوصلة A	الوصلة B	الوصلة C	الوصلة D
00 01	01 10	10 11	11 00

في إجابتك على الأسئلة التالية تذكَّر أن كلاً من الدوائر الافتراضية الحالية يمكن أن يمر خلال أحد الوصلات الأربعة فقط:

a. إذا كان من المطلوب أن تستخدم كل دائرة افتراضية نفس الرقم على كل الوصلات التي يتألف منها مسارها، فما رقم الدائرة الافتراضية الذي يمكن أن يخصص لدائرة افتراضية جديدة؟

b. إذا كان من المُصرَّح به أن تستخدم كل دائرة افتراضية أرقاماً مختلفة على الوصلات المختلفة على مسارها (لكي يلزم جداول التمرير أن تقوم بترجمة رقم الدائرة الافتراضية)، فما عدد التوافقات المختلفة للأربعة أرقام للدوائر الافتراضية (رقم لكلٍّ من الوصلات الأربعة) التي يمكن أن تُستخدَم؟

5. استخدمنا المصطلح "خدمة توصيلية" (connection-oriented service) لوصف خدمة طبقة النقل، واستخدمنا المصطلح "خدمة توصيلة" (connection service) لوصف خدمة طبقة الشبكة. ما سبب هذا التفريق الدقيق في المصطلحات؟

6. في الجزء 4-3 لاحظنا أنه قد لا يوجد صف انتظار عند المدخل إذا كان نسيج التحويل n مرة أسرع من معدلات خطوط الدخل (بافتراض وجود n خط دخل لكلٍّ منها نفس المعدل). وضح (بالوصف) سبب ذلك.

7. افترض موجّه بنسيج تحويل ومنفذي إدخال (A وB) ومنفذي إخراج (C وD). افترض أن سرعة نسيج التحويل 1.5 مرة سرعة الخط.

a. افترض (لسبب ما) أن كل الرزم من A متجهة إلى D، وكل الرزم من B متجهة إلى C. هل من الممكن أن يصمم نسيج تحويل بحيث لا يوجد صفوف انتظار لمنفذ الإدخال؟ وضح سبب موافقتك أو رفضك في جملة واحدة.

b. افترض الآن أن الرزم من A وB تتجه بشكلٍ عشوائي إلى C وD. هل من الممكن أن يصمم نسيج تحويل بحيث لا يوجد صفوف انتظار لمنفذ الإدخال؟ وضح سبب موافقتك أو رفضك في جملة واحدة.

8. افترض شبكة وحدات بيانات تستخدم 32 بتاً لعناوين المضيفات، وافترض موجّهاً بأربع وصلات مرقمة من 0 إلى 3 وأنه يلزم توجيه الرزم لواجهات الوصلات كما يلي:

مدى عناوين الوجهة	واجهة الوصلة
من 00000000 00000000 00000000 11100000 إلى 11111111 11111111 11111111 11100000	0
من 00000000 00000000 00000000 11100001 إلى 11111111 11111111 00000000 11100001	1
من 11111111 11111111 00000001 11100001 إلى 11111111 11111111 11111111 11100001	2
ما عدا ذلك	3

a. كوِّن جدول تمرير مؤلَّفاً من أربعة صفوف، ويستخدم قاعدة تطابق البادئة الأطول، ويرسل الرزم إلى واجهات الوصلات الصحيحة.

b. صف كيف يحدد جدول التمرير السابق (في جزء السؤال 8-a) واجهة الوصلة الملائمة لرزم البيانات لكلٍ من عناوين الوجهة التالية:

01010101 01010001 10010001 11001000
00111100 11000011 00000000 11100001
01110111 00010001 10000000 11100001

9. افرض أن شبكة وحدات البيانات تستخدم عناوين للمضيفات مؤلفة من 8 بتات. وافرض أن موجّهاً يستخدم قاعدة تطابق البادئة الأطول وله جدول التمرير التالي:

البادئة المطابقة	الواجهة
00	0
01	1
10	2
11	3

حدد المدى المناظر لعناوين مضيفات الوجهة وعدد العناوين في كل مدى، لكل وجهة من تلك الواجهات الأربع.

10. افرض أن شبكة وحدات البيانات تستخدم عناوين للمضيفات مؤلفة من 8 بتات. وافرض أن موجّهاً يستخدم قاعدة تطابق البادئة الأطول وله جدول التمرير التالي:

البادئة المطابقة	الواجهة
1	0
11	1
111	2
ما عدا ذلك	3

حدد المدى المناظر لعناوين مضيفات الوجهة وعدد العناوين في كل مدى، لكلٍ من تلك الواجهات الأربعة.

11. افرض أن موجّهاً يربط بين ثلاث شبكات فرعية: S1، S2، S3. وافرض أن كل شبكة من تلك الشبكات الفرعية يجب أن تستخدم البادئة 223.1.17/24. وافرض أيضاً أن الشبكة S1 يجب أن تدعم 125 واجهة، وأن كل شبكة من الشبكات S1 وS2 يجب أن تدعم 60 واجهة. اذكر ثلاثة عناوين شبكات (بالصيغة a.b.c.d/x) تحقق تلك القيود.

12. ذكرنا مثالاً لجدول تمرير في الجزء 4-2-2 (يستخدم قاعدة تطابق البادئة الأطول). أعد كتابة جدول التمرير هذا مستخدماً الصيغة a.b.c.d/x بدلاً من صيغة الأرقام الثنائية.
13. طُلِب منك في التمرين 7 إعطاء جدول تمرير (يستخدم قاعدة تطابق البادئة الأطول). أعد كتابة جدول التمرير هذا مستخدماً الصيغة a.b.c.d/x بدلاً من صيغة الأرقام الثنائية.
14. افرض أن شبكة فرعية تستخدم البادئة 101.101.101.64/26. أعط مثالاً لعنوان IP (بالصيغة xxx.xxx.xxx.xxx) يمكن تخصيصه لتلك الشبكة. افرض أن كتلة عناوين موفر خدمة الإنترنت لها الشكل 101.101.128/17، وأنه يريد تشكيل أربع شبكات فرعية من هذه الكتلة بكلٍّ منها نفس عدد عناوين IP. فما هي البادئات (بالصيغة a.b.c.d/x) للشبكات الفرعية الأربع؟
15. افرض أن شبكة لها الشكل الطبوغرافي المبين في الشكل 4-17. ولنرمز للشبكات الفرعية الثلاث التي فيها مضيفات (في اتجاه عقارب الساعة من الوضع 12:00) بـ A، B، C؛ للشبكات الفرعية بدون مضيفات بـ D، E، F.
    a. خصص عنوان شبكة لكلٍّ من تلك الشبكات الفرعية الست بحيث: يجب أن تخصص كل العناوين من 214.97.254/23، ويجب أن يكون للشبكة A عناوين تكفي لدعم 250 واجهة، ويجب أن يكون للشبكة B عناوين تكفي لدعم 120 واجهة، ويجب أن يكون للشبكة C عناوين تكفي لدعم 120 واجهة. بالطبع يجب أن تكون كل شبكة من الشبكات D وE وF قادرة على دعم واجهتين. ويجب أن تأخذ العناوين الصيغة a.b.c.d/x أو a.b.c.d/x – e.f.g.h/y.
    b. حدد جداول التمرير (طبقاً لقاعدة تطابق البادئة الأطول) لكلٍّ من الموجّهات الثلاثة، مستخدماً إجابتك السابقة على الجزء (a).
16. افرض أنك ترسل وحدة بيانات مكونة من 3000 بايت إلى وصلة لها الحد الأقصى لوحدة النقل MTU يعادل 500 بايت. افرض أن وحدة البيانات الأصلية مختومة بالرقم التعريفي 422. كم عدد الرزم الجزئية (fragments) المولّدة؟ وما خصائصها؟
17. افرض أن حجم وحدة البيانات بين مضيف المصدر A ومضيف الوجهة B لا يزيد عن 1500 بايت (بما في ذلك الترويسة). افرض أن حجم الترويسة 20 بايتاً، فكم عدد وحدات البيانات اللازمة لإرسال ملف MP3 حجمه 4 مليون بايت؟
18. افرض أن موفر خدمة الإنترنت خصص العنوان 126.13.89.67 لموجّه للشبكة الموجودة في الشكل 4-22، وأن عنوان شبكة البيت 192.168/16.
    a. خصص عناوين لكل الواجهات التي في شبكة البيت.

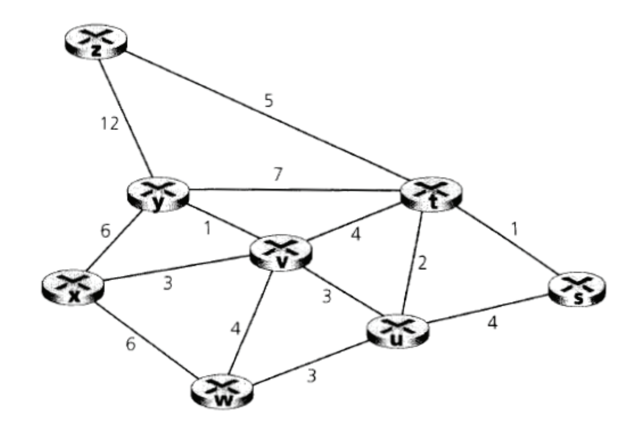
b. افرض أن كل مضيف له توصيلتان TCP وجميعها متجهة للمنفذ 80 على المضيف 128.119.40.86. حدد المدخلات الستة المناظرة في جدول ترجمة NAT.

19. في هذا التمرين سنفحص تأثير NAT على تطبيقات النظائر. افرض أن نظيراً باسم المستخدم آرنولد (Arnold) اكتشف من خلال الاستفسارات أن نظيراً آخر باسم المستخدم بيرنارد (Bernard) لديه ملف يريد تنزيله. وافرض أيضاً أن كلاً من بيرنارد وآرنولد وراء NAT. حاول ابتكار طريقة تسمح لآرنولد بتأسيس توصيلة TCP مع بيرنارد بدون إعداد NAT خاص بالتطبيق. إذا استصعب عليك الأمر لابتكار مثل هذه الطريقة، فناقش لماذا.

20. في الشكل 4-27 اذكر المسارات من v إلى y التي لا تتضمن حلقات (مسارات مغلقة).

21. أعد إجابة التمرين 20 للمسارات من x إلى w، ومن w إلى u، ومن z إلى x.

22. بالنظر إلى الشبكة التالية ذات كُلَف الوصلات المبينة، احسب أقصر مسار من x إلى كل عقدة من عقد الشبكة باستخدام خوارزمية Dijkstra. بيِّن خطوات الحل بإعداد جدول مشابه للجدول 4-3.



23. خذ في الاعتبار الشبكة المبينة بالتمرين 22، واستخدم خوارزمية Dijkstra مع بيان خطوات الحل (بإعداد جدول مشابه للجدول 4-3) لكلٍّ مما يلي:

a. احسب أقصر مسار من s إلى كل عقدة من عقد الشبكة.

b. احسب أقصر مسار من t إلى كل عقدة من عقد الشبكة.

c. احسب أقصر مسار من u إلى كل عقدة من عقد الشبكة.

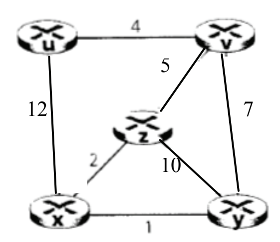
d. احسب أقصر مسار من v إلى كل عقدة من عقد الشبكة.

e. احسب أقصر مسار من w إلى كل عقدة من عقد الشبكة.

f. احسب أقصر مسار من y إلى كل عقدة من عقد الشبكة.

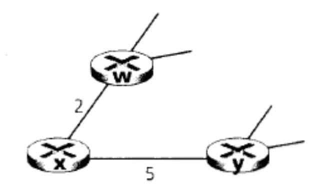
g. احسب أقصر مسار من z إلى كل عقدة من عقد الشبكة.

24. بفرض أن كل عقدة تعرف من البداية الكلفة إلى كلٍّ من جيرانها، استخدم خوارزمية متجه المسافة واحسب مدخلات جدول المسافات في العقدة z في الشبكة المبينة.



25. خذ في الاعتبار الشكل الطبوغرافي العام (أي ليس لشبكة معينة) واستخدم نسخة متزامنة من خوارزمية متجه المسافة. افرض أنه في كل تكرار تُرسل عقدة واحدة متجه المسافة لها إلى جيرانها وتستقبل متجهات المسافة من كلٍّ منهم. على فرض أنه عندما تبدأ الخوارزمية تعرِف كل عقدة الكُلَف إلى جيرانها المباشرين فقط، ما العدد الأقصى للتكرار اللازم لتقارب تلك الخوارزمية الموزّعة؟ بيّن السبب.

26. خذ في الاعتبار جزء الشبكة المبين بالشكل التالي، وفيه تتصل العقدة x بجارين w وy. افرض أن أدنى كلفة من w إلى الوجهة u تساوي 5، وأدنى كلفة من y إلى u تساوي 6. العقدة u والمسارات الكاملة من w وy إلى u (وبين w وy) غير مبينة بالشكل. افرض أيضاً أن جميع كلف الوصلات بالشبكة لها قيم صحيحة موجبة.



a. اذكر متجه المسافة لـ x للوجهات w وy وu.

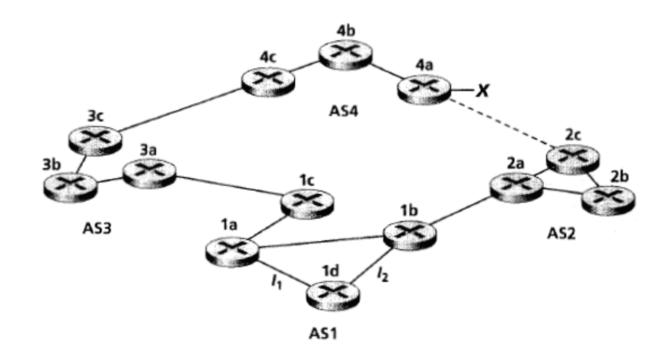
b. اذكر تغييراً في كلفة الوصلة $c(x, w)$ أو $c(x, y)$ ينتج عنه أن تخبر x جيرانها بمسار جديد للعقدة u كنتيجة لتنفيذ خوارزمية متجه المسافه.

c. اذكر تغييراً في كلفة الوصلة $c(x, w)$ أو $c(x, y)$ لا ينتج عنه أن تخبر x جيرانها بمسار جديد للعقدة u كنتيجة لتنفيذ خوارزمية متجه المسافه.

27. خذ في الاعتبار الطبوغرافية المبينة بالشكل 4-30 والمكونة من ثلاث عقد. لكن بدلاً من كُلَف الوصلات المبينة بالشكل افرض أن كلفة الوصلات كالتالي: $c(x,y) = 5$، $c(x,z) = 6$، $c(z,x) = 2$. احسب جداول المسافات بعد خطوة التهيئة وبعد كل خطوة من خطوات النسخة المتزامنة من خوارزمية متجه المسافة (كما فعلنا في المناقشة السابقة للشكل 4-30)

28. صف كيف يمكن الكشف عن وجود حلقات (مسارات مغلقة) في بروتوكول BGP.

29. خذ في الاعتبار الشبكة المبينة في الشكل التالي. افرض أن AS3 وAS2 يستخدمان بروتوكول OSPF، وأن AS1 وAS4 يَستخدمان بروتوكول RIP للتوجيه داخل النظام المستقل ذاتياً. افرض أن eBGP وiBGP يُستخدمان للتوجيه بين النظم المستقلة ذاتياً. في البداية افرض عدم وجود وصلة مادية بين AS2 وAS4.



a. من أي بروتوكول يعرف الموجّه 3c البادئة x هل هو OSPF أم RIP أم eBGP أم iBGP؟

b. من أي بروتوكول يعرف الموجّه 3a البادئة x؟

c. من أي بروتوكول يعرف الموجّه 1c البادئة x؟

d. من أي بروتوكول يعرف الموجّه 1d البادئة x؟

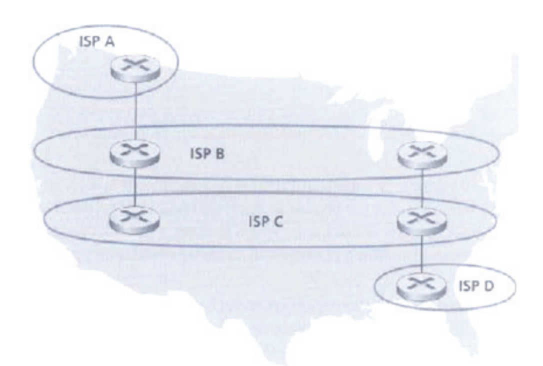
30. بالرجوع للتدريب السابق، بمجرد أن يعرف الموجّه 1d عن البادئة x سوف يضيف المُدخل (x, L) إلى جدول التمرير لديه.

a. هل تساوي L لهذا المُدخل l_1 أم l_2؟ وضح السبب في جملة واحدة.

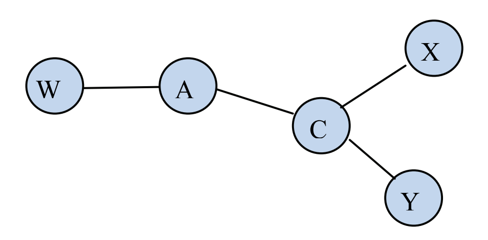
b. الآن افرض وجود وصلة مادية بين AS2 وAS4 والمبينة بالخط المنقوط. وافرض أن الموجّه 1d علم أنه يمكن الوصول لـ x عن طريق كل من AS2 وAS3. فهل تساوي L لهذا المُدخل l_1 أم l_2؟ وضح السبب في جملة واحدة.

c. الآن افرض وجود نظام مستقل ذاتياً آخر AS5 يقع على المسار بين AS2 وAS4 (غير مبين بالشكل). وافرض أن الموجّه 1d علم أنه يمكن الوصول لـ x عن طريق كلٍّ من AS2 AS5 AS4 وكذلك AS3 AS4. فهل تساوي L لهذا المُدخل l_1 أم l_2؟ وضح السبب في جملة واحدة.

31. خذ في الاعتبار الشبكة التالية. وفيها يوفر موفر خدمة الإنترنت B خدمة شبكة العمود الفقري القومي لموفر خدمة الإنترنت الإقليمي A، ويوفر موفر خدمة الإنترنت C خدمة شبكة العمود الفقري القومي لموفر خدمة الإنترنت الإقليمي D. بفرض أن كل موفر خدمة يتكون من نظام مستقل ذاتياً. يتصل B وC معاً من خلال وصلة ربط نظائر في موضعين باستخدام بروتوكول BGP. خذ في الاعتبار حركة مرور البيانات من A إلى D. سيفضل B أن يرسل حركة مرور البيانات إلى C من خلال وصلة الساحل الغربي (وبالتالي تتكلف C كلفة العبور خلال الولايات). في حين ستفضل C أن تستقبل حركة مرور البيانات من B من خلال وصلة الساحل الشرقي (وبالتالي تتكلف B كلفة العبور خلال الولايات). ما الآلية التي يستخدمها BGP لدى C حتى تقوم B بإرسال حركة مرور البيانات من خلال وصلة الساحل الشرقي؟ لكي تجيب على هذا السؤال ستحتاج للغوص في مواصفات BGP.



32. في الشكل 4-43 خذ في الاعتبار معلومات المسار التي تصل للشبكات الطرفية W وX وY. اعتماداً على المعلومات المتوفرة لكلٍّ من W وX، ما هو منظورهما لطبوغرافية الشبكة؟ علل إجابتك. مثلاً من منظور العقدة Y تكون طبوغرافية الشبكة كالتالي:



33. خذ في الاعتبار الشبكة المكونة من ثماني عقد (والمسماة بالحروف من s إلى z) في التمرين 21. بيّن الشجرة ذات أقل كلفة والتي جذرها s وتضم العقد u وv وw وy (كمضيفات طرفية). بشكلٍ تقريبي ناقش أسباب كون تلك الشجرة ذات أقل كلفة.
34. خذ في الاعتبار الطريقتين الأساسيتين اللتين ذكرناهما للإرسال الإذاعي: الأولى بمحاكاة الإرسال الفردي، والثانية بمعاونة الموجّه للإذاعة في طبقة الشبكة. افرض أننا استخدمنا الإرسال عبر الشجرة الممتدة للإذاعة في طبقة الشبكة، وافرض وجود مُرسِل وحيد و32 مُستقبِل، وأن المُرسِل موصل بالمُستقبلين عن طريق شجرة ثنائية من الموجّهات. ما كلفة إرسال رزمة إذاعة في كلٍّ من الحالتين (أي عند محاكاة الإرسال الفردي وعند الإذاعة في طبقة الشبكة)؟ هنا في كل مرة تُرسَل رزمة (أو نسخة منها) على وصلة تتكلف وحدة التكلفة. ما الشكل الطبوغرافي لربط المُرسِل والمستقبلين والموجّهات والذي يجعل كلفة كلٍّ من تلك الطريقتين بعيدة جداً بقدر الإمكان عن الأخرى؟ يمكنك أن تختار أي عدد من الموجّهات.
35. خذ في الاعتبار طريقة عمل خوارزمية تمرير المسار العكسي (RPF) في الشكل 4-45. باستخدام نفس الطبوغرافية حدد مجموعة المسارات من جميع العقد إلى عقدة المصدر A (ووضح تلك المسارات في رسم بياني مستخدماً خطوط سميكة مظلّلة كما في الشكل 4-45) بحيث إذا إذا كانت هذه المسارات تمثل المسارات ذات الكلفة الأقل، فعندئذ ستستقبل العقدة B نسخة من الرسالة المذاعة من العقدة A من العقد A وC وD.
36. خذ في الاعتبار الطبوغرافية المبينة في الشكل 4-45. افرض أن كلفة كل وصلة تعادل الوحدة وأن العقدة E هي مصدر الإذاعة. باستخدام أسهم كالمبينة بالشكل 4-45، وضح الوصلات التي سترسل الرزم خلالها باستخدام تمرير المسار العكسي (RPF)، وبيّن المسارات التي لن ترسل الرزم من خلالها.
37. خذ في الاعتبار الطبوغرافية المبينة في الشكل 4-47 وافترض أن كلفة كل وصلة تعادل الوحدة. افترض أن العقدة C اختيرت كمركز في خوارزمية توجيه متعدد

معتمدة على مركز (center-based multicast). على افتراض أن كل موجّه ملحق يستخدم مسار أقل كلفة إلى العقدة C لإرسال رسائل الالتحاق (join messages) إلى C، ارسم شجرة التوجيه الناتجة. هل تلك الشجرة الناتجة شجرة أدنى كلفة؟ اذكر أسباب إجابتك.

38. في الجزء 4-5-1 درسنا خوارزمية Dijkstra لتوجيه حالة الوصلة، والتي تقوم بحساب مسارات الإرسال الفردي (unicast) ذات أقل كلفة من المصدر إلى كلٍّ من الوجهات بشكلٍ مستقل. وتُشكِّل هذه المسارات مجتمعة شجرة مسارات أدنى كلفة (أو شجرة أقصر المسارات للإرسال الفردي إذا كانت كلف الوصلات متماثلة). بعرض مثال مضاد وضح أن شجرة مسارات أدنى كلفة ليست دائماً تماماً مثل الشجرة الممتدة بأدنى كلفة (minimum spanning tree).

39. خذ في الاعتبار شبكة فيها كل عقدة موصلة بثلاث عقد أخرى. في لحظة زمنية معينة يمكن أن تستقبل كل عقدة الرزم المذاعة المرسلة من جيرانها وتستنسخ تلك الرزم وترسلها إلى كلٍّ من جيرانها (ماعدا العقدة التي أرسلت تلك الرزمة). في اللحظة التالية يمكن أن تستقبل العقد المجاورة الرزم وتستنسخها وترسلها إلى جيرانهم، وهكذا. افترض أن الفيضان غير المحكوم يُستخدَم في الإرسال الإذاعي في مثل تلك الشبكة. في اللحظة t كم عدد النسخ للرزمة المذاعة ستُرسَل بافتراض أنه أثناء الخطوة الزمنية 1 تم إرسال رزمة مذاعة وحيدة من عقدة المصدر إلى جيرانها الثلاثة؟

40. رأينا في الجزء 4-7 أنه لا يوجد بروتوكول بطبقة الشبكة يمكن استخدامه لتحديد المضيفات المشترِكة في مجموعة إرسال متعدد. اشرح كيف تعرف تطبيقات الإرسال المتعدد هوية كلٍّ من تلك المضيفات المشترِكة بالمجموعة.

41. صمم (بوصف خطوات الإجراء) بروتوكولاً بطبقة التطبيقات يحتفظ بعناوين المضيفات المشترِكة في مجموعة إرسال متعدد. بالتحديد عيِّن نوع خدمة الشبكة التي يستخدمها هذا البروتوكول، هل هو إرسال فردي أو متعدد، ثم بين ما إذا كان هذا البروتوكول يرسل رسائله داخل النطاق (in-band) أو خارج النطاق (out-of-band) وذلك بالنسبة للبيانات المتدفقة من التطبيق بين المجموعة المشترِكة، مع بيان الأسباب.

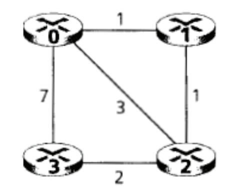
42. ما حجم فضاء عناوين الإرسال المتعدد؟ افرض الآن أن مجموعتين للإرسال المتعدد اختارتا عنواناً للإرسال المتعدد بشكلٍ عشوائي. ما احتمال اختيارهما لنفس العنوان؟ افرض أن 1000 مجموعة بدأت في نفس اللحظة اختيار العنوان بشكلٍ عشوائي، ما احتمال أن يحدث تداخل بين بعضهم البعض؟

❖ أسئلة للمناقشة

1. ابحث عن ثلاث شركات تبيع حالياً موجّهات بسرعات عالية، وقارن بين منتجاتها.
2. استخدم خدمة whois (من يكون؟) الموجودة في المسجِّل الأمريكي لأعداد الإنترنت (http://www.arin.net/whois) لتحديد كتل عناوين IP لثلاث جامعات. هل يمكن استخدام خدمة whois لتحديد بشكلٍ مؤكد الموقع الجغرافي لعنوان IP معين؟
3. هل من الممكن أن تكتب برنامج زبون ping (باستخدام رسائل ICMP) في لغة جافا؟ وضِّح السبب.
4. وضَّحنا في الجزء 4-4 أن انتشار بروتوكول IPv6 بطيء؛ بيّن أسباب ذلك؟ وما هو المطلوب لتسريع انتشاره؟
5. ناقش بعض مشاكل NAT المتعلقة بأمن IPSec (راجع كتاب [Phifer 2000])؟
6. قم بإعداد بحث عن بروتوكول UPnP. وبالتحديد صف الرسائل التي يستخدمها المضيف لإعادة تهيئة NAT.
7. افترض أن النظم المستقلة ذاتياً X وZ غير متصلة بشكلٍ مباشر وإنما متصلة عن طريق نظام آخر Y. وافترض أيضاً أن X لديه اتفاق نظائر (peering agreement) مع Y وأن Y لديه اتفاق نظائر مع Z. وأخيراً افترض أن Z يريد أن ينقل كل حركة بيانات Y ولا يريد أن ينقل حركة بيانات X. هل يسمح BGP لـ Z بتطبيق هذه السياسة؟
8. حددنا في الجزء 7-4 عدداً من تطبيقات الإرسال المتعدد (multicast)؛ أي من تلك التطبيقات يناسب بشكلٍ جيد نموذج خدمة الإرسال المتعدد في الإنترنت؟ وأي تطبيقات لا تناسب بشكلٍ جيد تحديداً نموذج الخدمة هذا؟ بين السبب.

❖ تدريبات على برمجة المقابس

في هذا التدريب ستكتب مجموعة من الإجراءات الموزّعة لتحقيق توجيه بمتجه المسافة لاتزامني موزّع للشبكة التالية:



يمكنك الحصول على التفاصيل الكاملة لهذا التمرين وكذلك أجزاء من البرنامج في لغة C وأيضاً في لغة جافا من خلال موقع الويب لهذا الكتاب http://www.awl.com/kurose-ross.

❖ تدريبات معملية على استخدام برنامج Ethereal

في موقع الويب الخاص بهذا الكتاب http://www.awl.com/kurose-ross ستجد تدريبين على استخدام برنامج Ethereal لهذا الفصل. يفحص التدريب الأول طريقة عمل بروتوكول IP وبشكلٍ محدد صيغة وحدة بيانات IP. أما التدريب الثاني فيفحص استخدام بروتوكول ICMP في أوامر ping (لاختبار اتصال الأجهزة المختلفة بالشبكة) وtraceroute (لتتبع المسارات إلى الأجهزة المختلفة).

طبقة ربط البيانات والشبكات المحلية

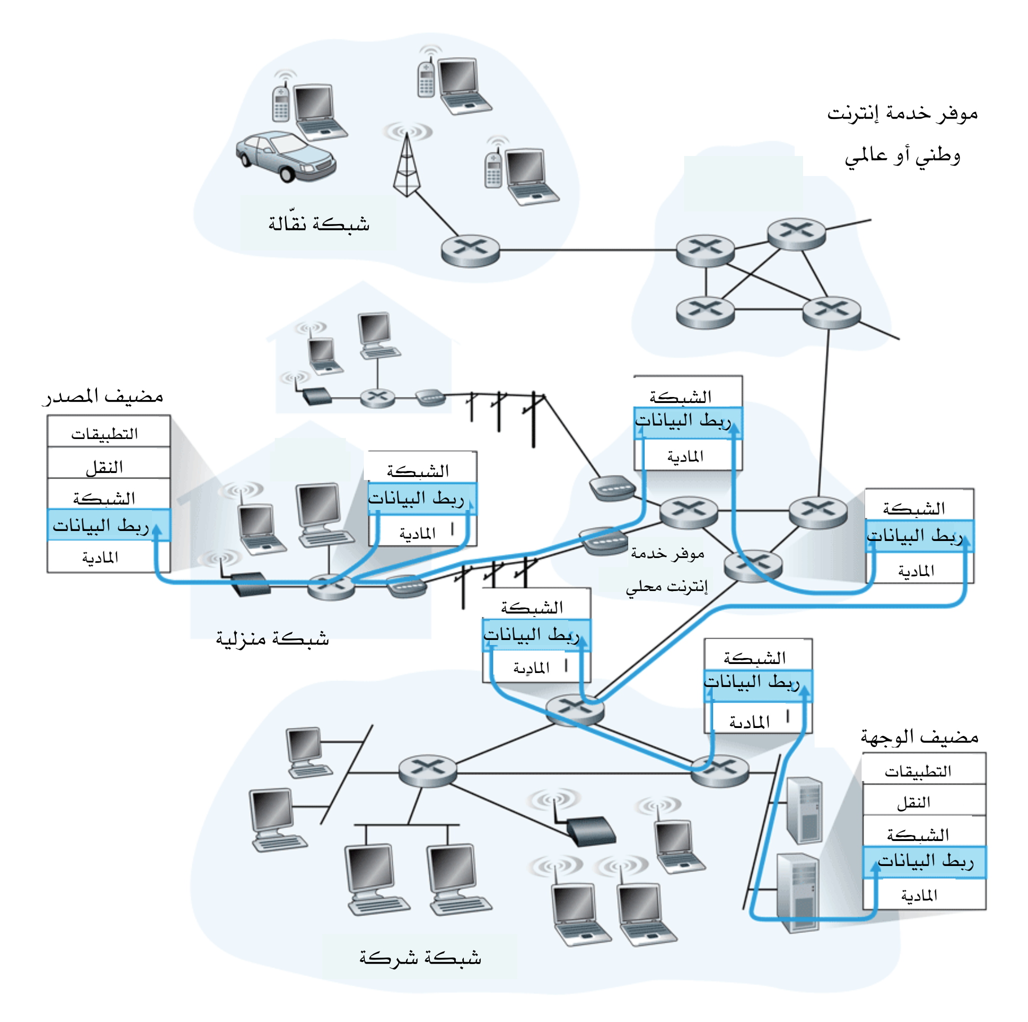
The Link Layer and Local Area Networks

محتويات الفصل:

- مقدمة عن طبقة ربط البيانات وخدماتها
- أساليب اكتشاف أخطاء البيانات وتصحيحها
- بروتوكولات الوصول المتعدد
- العنونة في طبقة ربط البيانات
- شبكة الإيثرنت
- محوّلات طبقة ربط البيانات
- بروتوكول نقطة إلى نقطة (PPP)
- الوصلة الافتراضية: الشبكة كطبقة ربط البيانات
- الخلاصة

في الفصل السابق عرفنا كيف تقوم طبقة الشبكة بتوفير خدمة اتصال بين مضيفين. كما هو موضح في الشكل 5-1 يتألف مسار الاتصال من سلسلة من الوصلات تبدأ من مضيف المصدر وتمر بسلسلة من الموجّهات حتى تنتهي إلى مضيف الوجهة. بينما نواصل المضي إلى أسفل عبر رصة البروتوكولات (أي من طبقة الشبكة إلى طبقة ربط البيانات)، من الطبيعي أن نتساءل كيف تُرسل الرزم عبر الوصلات المختلفة التي تكوّن مسار الاتصال من طرف إلى طرف؟ كيف يتم تغليف وحدات بيانات طبقة الشبكة في إطارات طبقة ربط البيانات تمهيداً لإرسالها على وصلة واحدة؟ هل بإمكان بروتوكولات طبقة ربط البيانات توفير نقل موثوق للبيانات من موجّه إلى موجّه؟ هل يمكن استخدام بروتوكولات مختلفة لطبقة ربط البيانات على طول مسار اتصالٍ ما؟ سنجيب على هذه الأسئلة وغيرها من الأسئلة المهمة في هذا الفصل.

في دراستنا لطبقة ربط البيانات سنجد أن هناك نوعين مختلفين بشكل جوهري من قنوات طبقة ربط البيانات. يضم النوع الأول قنوات الإذاعة (broadcast channels)، والتي توجد بكثرة في الشبكات المحلية (LANs)، والشبكات المحلية اللاسلكية، وشبكات الأقمار الصناعية، والشبكات الهجينة ذات الألياف الضوئية والكبلات المحورية (HFC). في هذا النوع من قنوات الإذاعة يوصّل العديد من المضيفات بنفس قناة الاتصال، ومن ثم يحتاج الأمر إلى ما يسمّى ببروتوكول الوصول للوسط لتنسيق عملية الإرسال على تلك القناة المشتركة ولتفادي الاصطدام بين الإطارات المُرسَلة. أما النوع الثاني من قنوات طبقة ربط البيانات فهو وصلة الاتصال من نقطة إلى نقطة، كما هو الحال بين موجّهين أو بين المودم والموجّه الخاص بموفر خدمة الإنترنت. واضح أن تنسيق الوصول إلى وصلة من نوع نقطة إلى نقطة تعتبر عملية سهلة، إلا أنه لا يزال هناك عدد من القضايا الهامة تتعلق بالتأطير، والنقل الموثوق للبيانات، واكتشاف الأخطاء، وضبط التدفق.



الشكل 5-1 طبقة ربط البيانات.

سنستكشف في هذا الفصل عدّة تقنيات مهمة لطبقة ربط البيانات. سنلقي نظرة متعمقة على شبكة الإيثرنت، والتي تعتبر إلى حد كبير التقنية الأكثر رواجاً وأهمية للشبكات المحلية، وسنتناول كذلك بروتوكول نقطة إلى نقطة (PPP)، وهو البروتوكول المفضل الذي تستخدمه المضيفات السكنية في الوصول للإنترنت عن طريق المودم.

رغم أن شبكة WiFi – والشبكات المحلية اللاسلكية على وجه العموم – تُعد بالتأكيد مواضيع ضمن طبقة ربط البيانات، إلا أننا لن نغطيها في هذا الفصل. ليس ذلك لعدم أهميتها، فثورة WiFi تغيّر بشكلٍ مثير الطريقة التي يدخل بها الناس على الإنترنت ويستعملونها، ولكننا سنغطي هذا الموضوع بعمق في الفصل السادس، والذي يركز على شبكات الحاسب اللاسلكية وقابلية الحركة.

5-1 مقدمة عن طبقة ربط البيانات وخدماتها

دعنا نبدأ ببعض المصطلحات المفيدة. ونرى أنه من الأسهل في هذا الفصل الإشارة إلى المضيفات والموجّهات ببساطة كعقد، حيث لن نكترث بشكلٍ خاص ـ كما سنبين بعد قليل ـ بما إذا كانت تلك العقدة مضيفاً أو موجّهاً. سنشير أيضاً إلى قنوات الاتصال التي تربط ما بين العقد المتجاورة على طول مسار الاتصال كوصلات. لكي يتم نقل وحدة بيانات من مضيف المصدر إلى مضيف الوجهة يجب أن تنتقل عبر كل الوصلات المختلفة كلٍّ على حدة على المسار من طرف إلى طرف. وعلى كل وصلة تقوم العقدة (التي عند أحد طرفي الوصلة) بتغليف وحدة البيانات في إطار طبقة ربط البيانات ثم ترسل الإطار عبر تلك الوصلة . تتلقّى عقدة الاستلام الإطار عند طرف الوصلة الآخر، ثم تقوم بدورها بانتزاع وحدة البيانات منه.

5-1-1 الخدمات التي توفرها طبقة ربط البيانات

يُستخدم بروتوكول طبقة ربط البيانات لنقل وحدات بيانات طبقة الشبكة على نفس الوصلة. يعرّف بروتوكول طبقة ربط البيانات صيغة الرزم التي يتم تبادلها بين العُقد الموجودة عند طرفي الوصلة، وكذلك الإجراءات التي تتخذها تلك العُقد عند إرسال واستلام الرزم. تذكّر أنه مرّ علينا في الفصل الأول أن رزم البيانات التي يتم تبادلها ضمن بروتوكول طبقة ربط البيانات يُطلق عليها إطارات، وأن كل إطار من إطارات طبقة ربط البيانات يغلّف في العادة وحدة بيانات لطبقة الشبكة. كما سنرى بعد قليل تتضمن الأعمال التي يقوم بها بروتوكول طبقة ربط البيانات عند إرسال واستلام الإطارات: اكتشاف أخطاء البيانات، وإعادة الإرسال، وضبط

التدفق، والوصول العشوائي. من أمثلة بروتوكولات طبقة ربط البيانات: بروتوكول الإيثرنت للشبكات المحلية، وبروتوكول 802.11 للشبكات المحلية اللاسلكية (والمعروفة أيضاً بـ WiFi)، وبروتوكول حلقة العلامة (token ring)، وبروتوكول نقطة إلى نقطة (PPP). وفي العديد من السياقات يمكن اعتبار بروتوكول نمط النقل غير المتزامن (ATM) بروتوكول طبقة وصلة أيضاً. سنغطي العديد من هذه البروتوكولات بالتفصيل في النصف الثاني من هذا الفصل.

في حين تضطلع طبقة الشبكة بمهمة نقل قطع بيانات طبقة النقل من طرف إلى طرف (أي من مضيف المصدر إلى مضيف الوجهة) عبر مسار الاتصال، تقتصر مهمة بروتوكول طبقة ربط البيانات على نقل وحدات بيانات طبقة الشبكة عبر كل وصلة على حدة من ذلك المسار (أي من عقدة إلى عقدة). من الخصائص المهمة لطبقة ربط البيانات أن وحدة البيانات يمكن أن تُنقل ببروتوكولات مختلفة لطبقة ربط البيانات على الوصلات المختلفة عبر المسار. فمثلاً قد تُنقل وحدة بيانات ببروتوكول الإيثرنت على الوصلة الأولى، وببروتوكول نقطة إلى نقطة (PPP) على الوصلة الأخيرة، وببروتوكول شبكة المناطق الواسعة (WAN) على الوصلات المتوسّطة. من المهم أيضاً ملاحظة أن البروتوكولات المختلفة لطبقة ربط البيانات قد توفر خدمات مختلفة. فعلى سبيل المثال توفر بعض تلك البروتوكولات توصيلاً موثوقاً للبيانات في حين لا توفر بروتوكولات أخرى ذلك، ولهذا يتعين أن تكون طبقة الشبكة قادرة على تحقيق مهمتها من طرفٍ إلى طرف في وجود مجموعة متباينة من خدمات طبقة ربط البيانات على الوصلات الفردية التي تكوّن مسار الاتصال الكلي.

لكي نفهم طبيعة طبقة ربط البيانات وعلاقتها بطبقة الشبكة، دعنا نتأمل المثال التالي من عالم السفريات. افترض أن وكيل سفريات يخطط لرحلة لأحد السياح من برينستون في نيو جيرسي إلى لوزان بسويسرا. افترض أنه قرر أنه من الملائم للسائح أن يركب سيارة من برينستون إلى مطار JFK بنيويورك، ثم طائرة من مطار JFK إلى مطار جنيف، وأخيراً قطاراً من مطار جنيف إلى محطة قطار لوزان. بعد قيام الوكيل بعمل الحجوزات الثلاثة، تكون مسؤولية شركة ليموزين

برينستون توصيل السائح من برينستون إلى JFK، ومسؤولية شركة الطيران توصيل السائح من JFK إلى جنيف، ومسؤولية مصلحة القطارات السويسرية توصيل السائح من جنيف إلى لوزان. تمثل كل مرحلة من الرحلة تلك انتقالاً "مباشراً" بين موقعين "متجاورين"، كما أن تلك المراحل تدار من قِبَل شركات مختلفة وتستخدم وسائل انتقال مختلفة تماماً (ليموزين، وطائرة، وقطار). ومع ذلك وبالرغم من أن أنماط النقل مختلفة، فإنها تقدم الخدمة الأساسية لنقل المسافرين من موقع إلى موقع آخر مجاور. في هذا المثال من عالم السفريات، يمثل السائح "وحدة بيانات"، بينما تمثل كل مرحلة من الرحلة "وصلة اتصال"، وكل نمط نقل يمثل "بروتوكول طبقة ربط البيانات"، في حين يقوم وكيل السفريات بدور "بروتوكول التوجيه".

رغم أن الخدمة الأساسية لطبقة ربط البيانات تنحصر في نقل وحدة البيانات من عقدة إلى عقدة مجاورة على وصلة اتصال واحدة، فإن تفاصيل الخدمة التي يتم توفيرها قد تتفاوت من بروتوكول لآخر في طبقة ربط البيانات. يمكن أن تتضمن الخدمات التي يوفرها بروتوكول طبقة ربط البيانات ما يلي:

- التأطير: تقوم كل بروتوكولات طبقة ربط البيانات تقريباً بتأطير كل وحدة بيانات من طبقة الشبكة ضمن إطار طبقة ربط البيانات قبل إرسالها على الوصلة. يتضمن الإطار حقل بيانات يتم فيه إدخال وحدة بيانات طبقة الشبكة، وعدداً من حقول الترويسة (header) في بدايته (ويمكن أن يتضمن الإطار حقولاً في نهايته أيضاً، غير أننا سنشير إلى كل هذه الحقول مجتمعةً على أنها "حقول الترويسة"). يحدد بروتوكول طبقة ربط البيانات هيكل الإطار، وسنرى عدة صيغ مختلفة للإطارات عند دراسة أمثلة محددة لبروتوكولات طبقة ربط البيانات في النصف الثاني من هذا الفصل.

- تنسيق الوصول للوصلة: يحدد بروتوكول التحكم في الوصول للوسط (medium access control (MAC)) القواعد التي تحكم عملية إرسال إطار على الوصلة. في حالة الوصلات من نقطة إلى نقطة حيث يوجد مُرسِل واحد على أحد طرفي الوصلة ومُستقبِل واحد على الطرف الآخر، يكون

بروتوكول MAC بسيطاً (أو غير موجود بالمرة) حيث يكون بوسع المُرسِل إرسال إطار في أي وقت تكون الوصلة فيه شاغرة (غير مستخدمة). أما الحالة الأكثر تشويقاً فهي عندما تشترك عدة عقد في وصلة إذاعة واحدة حيث نواجه مشكلة تُعرف بالوصول المتعدد للوصلة. في هذه الحالة يوفر بروتوكول MAC خدمة تنسيق عملية إرسال الإطارات من العديد من العقد، وسنغطي بروتوكولات MAC بالتفصيل في الجزء 5-3.

- التوصيل الموثوق للبيانات: عندما يوفر بروتوكول طبقة ربط البيانات خدمة توصيل موثوق، فإنه يضمن نقل كل وحدة بيانات لطبقة الشبكة عبر الوصلة بدون أخطاء. ولعلك تذكر أن بعض بروتوكولات طبقة النقل (كبروتوكول التحكم في الإرسال TCP) توفر هي الأخرى خدمة توصيل موثوق. كما في خدمة النقل الموثوق للبيانات بطبقة النقل، يتم توفير خدمة التوصيل الموثوق على مستوى طبقة ربط البيانات في أغلب الأحيان باستخدام إشعارات الاستلام وإعادة الإرسال (انظر الجزء 3-4). غالباً ما تستخدم خدمة طبقة ربط البيانات للتوصيل الموثوق على الوصلات التي تكون عرضة لمعدلات خطأ عالية في البيانات - كوصلات اللاسلكي - بهدف تصحيح الأخطاء التي تقع محلياً على الوصلة التى يحدث عليها الخطأ بدلاً من الالتجاء إلى إعادة إرسال البيانات من طرف إلى طرف عن طريق بروتوكولات طبقة النقل أو طبقة التطبيقات. ومع ذلك، فإن التوصيل الموثوق للبيانات على مستوى طبقة ربط البيانات قد يُعتبر عبئاً غير ضروري على الوصلات التي تمتاز بمعدلات خطأ منخفضة كالألياف الضوئية، والكبل المحوري، والعديد من وصلات أزواج الأسلاك النحاسية المجدولة. لهذا السبب فإن العديد من بروتوكولات طبقة ربط البيانات المستخدمة على الوصلات السلكية لا توفر خدمة توصيل موثوق بها.

- ضبط التدفق: تتوافر بالعقدتين على طرفي الوصلة سعة محدودة للتخزين المؤقت للإطارات، مما قد يؤدي إلى مشاكل محتملة. فقد تتلقى عقدة الاستقبال الإطارات بمعدل أسرع من المعدل الذي تستطيع معالجتها به، ولذا

قد يفيض المخزن المؤقت لدى المُستقبِل عند عدم توفر ضبط للتدفق مما يؤدي إلى فقد إطارات. كما هو الحال في طبقة النقل، يمكن أن يوفر بروتوكول طبقة ربط البيانات ضبطاً للتدفق لمنع عقدة الإرسال على أحد طرفي الوصلة من غمر عقدة الاستقبال على الطرف الآخر.

- اكتشاف الأخطاء: يمكن أن تقرر عقدة الاستقبال بشكلٍ خاطئ أن البت الذي استقبلته "0" بينما البت المُرسِل هو في الحقيقة "1"، والعكس بالعكس. تنشأ أخطاء البتات نتيجة اضمحلال الإشارة المُرسَلة واختلاطها بالضوضاء الكهرومغناطيسية. نظراً لأنه لا طائل من تمرير وحدة بيانات تتضمن خطأ في بتاتها، توفر العديد من بروتوكولات طبقة ربط البيانات آليات تمكّن المُستقبِل من اكتشاف وجود خطأ في بت واحد أو أكثر. لتحقيق ذلك تضيف عقدة الإرسال في الإطار مجموعة بتات خاصة باكتشاف الأخطاء، وفي المقابل تقوم عقدة الاستقبال بعملية فحص لاكتشاف وجود خطأ من عدمه. تعتبر آليات اكتشاف الأخطاء من الإمكانات المشهورة جداً في بروتوكولات طبقة ربط البيانات. ذكرنا في الفصلين الثالث والرابع أن كلاً من طبقتي النقل والشبكة توفر أيضاً إمكانيات محدودة لاكتشاف الأخطاء، كما في حالة المجموع التدقيقي (checksum) بالإنترنت. عادةً ما تكون وسائل اكتشاف الأخطاء في طبقة ربط البيانات أكثر تطوراً ويتم إنجازها في مكونات مادية (hardware) وليست برمجية (software).

- تصحيح الأخطاء: تشبه وسائل تصحيح الأخطاء وسائل اكتشاف الأخطاء، غير أن عقدة الاستقبال هنا لا تكتفي فقط باكتشاف ما إذا كانت هناك أخطاء في البتات قد طرأت على الإطار أثناء انتقاله ولكنها أيضاً تحدد بالضبط مواضع تلك الأخطاء في الإطار (ومن ثم يمكنها تصحيحها). بعض البروتوكولات (كبروتوكول نمط النقل غير المتزامن (ATM) توفر إمكانية لتصحيح الأخطاء في ترويسة الرزمة فقط وليس في الرزمة بكاملها. سنتناول اكتشاف وتصحيح الأخطاء في الجزء 5-2.

- اتصال مزدوج الإرسال في الاتجاهين (نصفي Half-duplex أو كامل Full-duplex): في حالة الإرسال المزدوج الكامل يمكن للعقدتين على طرفي الوصلة إرسال الرزم في نفس الوقت، وفي حالة الإرسال المزدوج النصفي لا تستطيع العقدة القيام بكلٍ من البث والاستقبال في نفس الوقت.

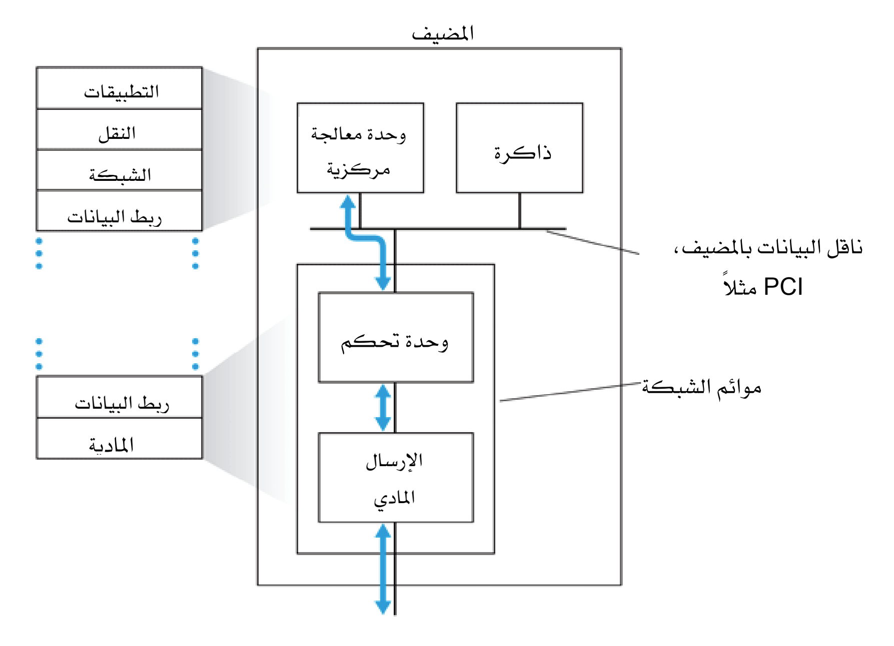
يتضح مما تقدم أعلاه أوجه الشبه القوية بين العديد من الخدمات التي توفرها طبقة ربط البيانات ونظيراتها من خدمات طبقة النقل. على سبيل المثال بوسع كلتا الطبقتين توفير توصيل موثوق للبيانات. ورغم تماثل الآليات المستخدمة في الطبقتين للحصول على ذلك التوصيل الموثوق (انظر الجزء 3-4)، فإن خدمتي التوصيل الموثوق في الحالتين ليستا واحدة. فبروتوكول النقل يوفر توصيلاً موثوقاً بين عمليتين على أساس من طرف إلى طرف، في حين يوفر بروتوكول طبقة ربط البيانات تلك الخدمة بين عقدتين متصلتين بوصلة واحدة. بنفس الطريقة يمكن أن توفر كلتا الطبقتين خدمات لضبط التدفق واكتشاف الأخطاء، ولكن - مرةً أخرى - يوفر بروتوكول النقل ضبط التدفق على أساس من طرف إلى طرف بينما يوفر بروتوكول طبقة ربط البيانات ذلك على أساس من عقدة إلى عقدة مجاورة فقط.

5-1-2 أين يُنفّذ بروتوكول طبقة ربط البيانات؟

قبل الخوض في دراستنا التفصيلية لطبقة ربط البيانات دعنا ننظر في مسألة المكان الذي يتم فيه إنجاز الوظائف والمهام المنوطة بتلك الطبقة. سنركّز هنا على نظام طرفي (حيث عرفنا في الفصل الرابع كيف تُضمّن وظائف طبقة ربط البيانات على الموجّه في بطاقة (كرت) الخط (line card))، فهل يتم إنجاز طبقة ربط البيانات على مضيف بواسطة مكونات مادية أم برمجية؟ هل تنجز على كرت أو رقاقة مستقلة؟، وكيف تتواصل مع بقيّة المكونات المادية للمضيف والأجزاء المختلفة لنظام التشغيل؟

يبين الشكل 5-2 مخططاً لبنية معمارية نمطية لمضيف. يتم إنجاز الجزء الأكبر من طبقة ربط البيانات في بطاقة التوصيل بالشبكة، والتي تُعرف أحياناً ببطاقة واجهة الشبكة (Network Interface Card (NIC)). تقبع وحدة التحكم

التي تؤدي مهام طبقة ربط البيانات في قلب بطاقة التوصيل بالشبكة، وعادةً ما تأخذ شكل رقاقة واحدة مصممة خصيصاً للقيام بالعديد من وظائف طبقة ربط البيانات (التأطير، الوصول للوصلة، ضبط التدفق، اكتشاف الأخطاء، إلخ)، والتي تعرّفنا عليها في الجزء السابق. وعليه فإن الجزء الأكبر من وظائف وحدة التحكم الخاصة بطبقة ربط البيانات يتم تنفيذها داخل مكونات مادية.



الشكل 5-2 بطاقة مواءمة الشبكة: علاقتها بمكونات المضيف الأخرى وبوظائف رصة البروتوكولات.

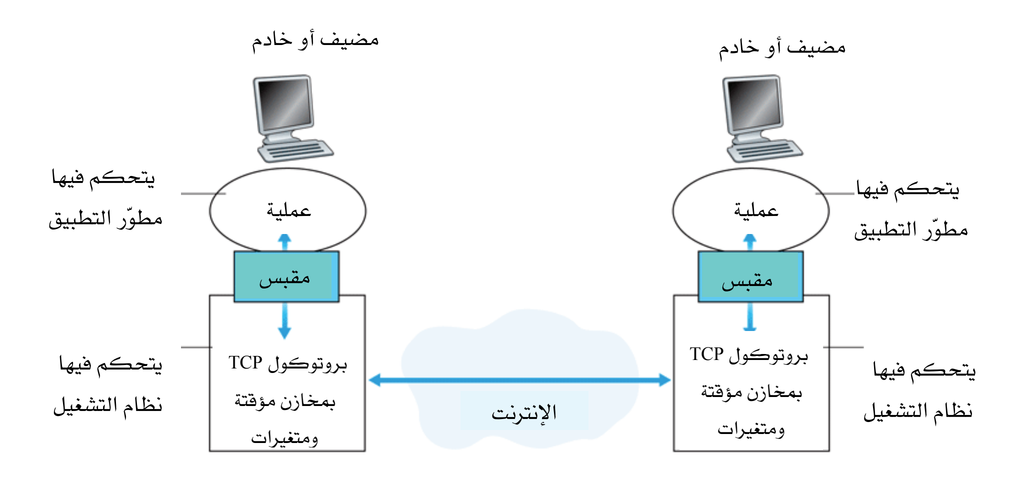
على سبيل المثال تحقق وحدة التحكم طراز 8254x من Intel [Intel 2006] بروتوكولات الإيثرنت التي سندرسها في الجزء 5-5، بينما تحقق وحدة التحكم طراز AR5006 من Atheros [Atheros 2006] بروتوكولات 802.11 WiFi التي سندرسها في الجزء 6-3. حتى أواخر التسعينيات كانت أكثر بطاقات المواءمة للشبكة مستقلةً مادياً (مثل بطاقة PCMCIA)، أو بطاقات تركّب في فتحة من فتحات التوسع القياسية في الحاسب الشخصي من نوع PCI مثلاً). أما الآن فيتم دمج

عددٍ متزايدٍ من بطاقات المواءمة للشبكة على اللوحة الأم (motherboard) للمضيف، ومن ثم الحصول على ترتيبة يطلق عليها LAN-on-motherboard (شبكة محلية على اللوحة الأم).

على جانب الإرسال: تأخذ وحدة التحكم رزمة البيانات التي أنشأتها الطبقات الأعلى من رصة البروتوكولات وخزّنتها في ذاكرة المضيف، وتغلفها في إطار طبقة ربط البيانات (بملء حقول الإطار المختلفة) ثم تقوم بعد ذلك ببث الإطار على الوصلة تبعاً للبروتوكول المستخدم للوصول للوصلة. على جانب الاستقبال: تستلم وحدة التحكم إطار طبقة ربط البيانات كاملاً، وتستخرج منه رزمة بيانات طبقة الشبكة. إذا كانت طبقة ربط البيانات تؤدي وظيفة اكتشاف الأخطاء، فإن وحدة التحكم على المُرسِل هي التي تضيف بتات اكتشاف الأخطاء في ترويسة الإطار، بينما تقوم وحدة التحكم على المُستقبِل بعملية الفحص لاكتشاف الأخطاء. أما إذا كانت طبقة ربط البيانات تتضمن ضبطاً للتدفق، فإن وحدتي التحكم على كلٍّ من المُرسِل والمُستقبِل تتبادلان رسائل تتضمن معلومات خاصة بضبط التدفق بحيث يقوم المُرسِل بإرسال الإطارات بمعدل يستطيع المُستقبِل التعامل معه.

يبين الشكل 5-2 بطاقة مواءمة للشبكة موصلة بناقل البيانات على مضيف (مثلاً ناقل البيانات (data bus) من نوع PCI أو PCI-X) حيث تبدو للمكونات الأخرى للحاسب المضيف كأي أداة أخرى على الناقل لإدخال وإخراج البيانات. كما يبين الشكل 5-2 أيضاً أنه رغم أن معظم وظائف بروتوكول طبقة ربط البيانات يتم إنجازها على مكونات مادية على بطاقة المواءمة إلا أن جزءاً من ذلك البروتوكول يجري تنفيذه من خلال برمجيات يتم تشغيلها على وحدة المعالجة المركزية للمضيف. تقوم الأجزاء البرمجية من طبقة ربط البيانات عادةً بتنفيذ الوظائف بالمستوى الأعلى من الطبقة كاستلام وحدة البيانات من طبقة الشبكة، وتجميع معلومات العنونة الخاصة بطبقة ربط البيانات، وتفعيل وحدة التحكم. على جانب المُستقبِل: تستجيب برمجيات طبقة ربط البيانات لإشارات المقاطعة (interrupts) التي تولدها وحدة التحكم (مثلاً عند استلام إطار أو أكثر) حيث تقوم بالتعامل مع حالات حدوث الخطأ، وتقوم بتمرير وحدة البيانات التي يتم

استلامها إلى طبقة الشبكة. وهكذا فإن طبقة ربط البيانات تمثل تشكيلة من المكونات المادية والبرمجية؛ إنها بمثابة المكان في رصة البروتوكول الذي تلتقي فيه المكونات المادية بالبرمجيات. يتضمن المرجع [Intel 2006] نظرةً عامّةً سهلة القراءة (مع وصف تفصيلي) لوحدة التحكم Intel طراز 8254 x من وجهة نظر برمجية.



الشكل 5-3 الاتصال بين بطاقات المواءمة: تغلّف رزمة بيانات طبقة الشبكة ضمن إطار طبقة ربط البيانات قبل إرسالها على الطبقة المادية.

يبين الشكل 5-3 بطاقات المواءمة للمُرسِل والمُستقبِل. لمّا كانت الوظائف الرئيسة لبروتوكول طبقة ربط البيانات تقوم بها وحدة التحكم، فإن بطاقات المواءمة تعتبر وحدات شبه ذاتية وظيفتها نقل إطار من بطاقة إلى أخرى. قام عدد من الباحثين بدراسة إمكانية نقل المزيد من الوظائف الأخرى (فيما وراء معالجة طبقة ربط البيانات) إلى بطاقات المواءمة للشبكة. فمثلاً بوسع وحدة التحكم طراز 8254x حساب المجموع التدقيقي (checksum) لقطع بيانات TCP/UDP ولترويسة وحدة بيانات IP – أي استخدام المكونات المادية (وحدة التحكم بطبقة ربط البيانات) لأداء وظائف تتبع طبقتي الشبكة والنقل. رغم أن هذا قد يبدو انتهاكاً صارخاً لمبدأ طبقية رصة البروتوكولات إلا أنه يحقق فائدة، فالمكونات المادية

يمكنها حساب المجموع التدقيقي أسرع بكثير من البرامج. يتضمن المرجع [Mogul 2003] مناقشة شائقة لفوائد ومضار القيام بعمليات المعالجة الخاصة ببروتوكول التحكم في الإرسال (TCP) على بطاقات المواءمة.

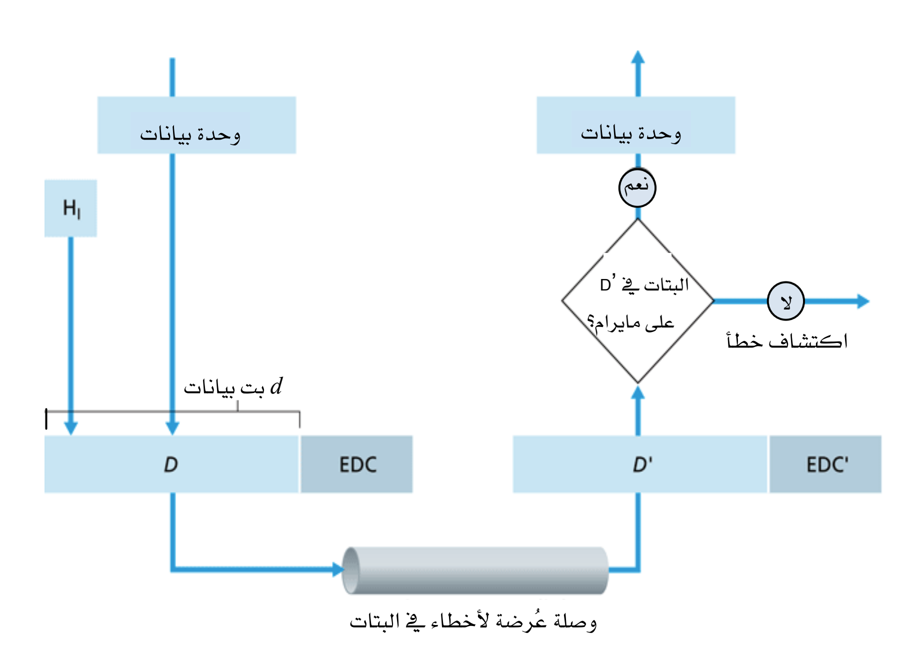
أما [Kim 2005] فينظر في إمكانية أداء وظائف طبقات أعلى حتى من ذلك على بطاقات المواءمة (مثل HTTP caching أي تخزين نسخة من ملفات HTTP المستخدمة بكثرة في الذاكرة المخبأة).

2-5 أساليب اكتشاف أخطاء البيانات وتصحيحها

في الفصل السابق ذكرنا أن اكتشاف وتصحيح أخطاء البيانات على مستوى البتات هما خدمتان توفرهما غالباً طبقة ربط البيانات، وذلك لاكتشاف وإصلاح أخطاء البتات التي يتكون منها إطار طبقة ربط البيانات أثناء انتقاله من عقدة إلى عقدة أخرى مجاورة تتصل بها عبر وسط مادي. رأينا في الفصل الثالث أن خدمات اكتشاف الأخطاء وتصحيحها يتم توفيرها في أغلب الأحيان في طبقة النقل أيضاً. في هذا الجزء سندرس بعض الأساليب البسيطة المستخدمة لاكتشاف – وفي بعض الأحيان تصحيح – مثل تلك الأخطاء. هناك العديد من الكتب الدراسية المخصصة لمعالجة نظرية وتطبيق هذا الموضوع بالكامل (على سبيل المثال [Schwartz 1980] أو [Bertsekas 1991]). ستكون معالجتنا للموضوع هنا مختصرة بالضرورة حيث نهدف لتطوير مفهوم بدهي للإمكانات التي توفرها أساليب اكتشاف وتصحيح الأخطاء، وتوضيح كيف تفي بعض الأساليب البسيطة بهذا الغرض وتُستخدم فعلياً في طبقة ربط البيانات.

يوضح الشكل 5-4 الإطار العام لدراستنا للموضوع. في عقدة الإرسال يُلحَق بالبيانات D المعدّة للإرسال والمطلوب حمايتها من تأثير الأخطاء مجموعة بتات خاصة باكتشاف وتصحيح الأخطاء (EDC). لا تقتصر البيانات المطلوب حمايتها عادةً على وحدة البيانات التي دفعت بها طبقة الشبكة لنقلها عبر الوصلة، ولكنها تتضمن أيضاً معلومات العنونة والأرقام التسلسلية والحقول الأخرى في ترويسة إطار بيانات طبقة ربط البيانات. يتم إرسال كلٍّ من D وEDC إلى عقدة الاستقبال ضمن

إطار طبقة ربط البيانات. في عقدة الاستقبال يتم استلام D' و'EDC. لاحظ أن كلاً من D' و'EDC قد يختلفان عن البيانات الأصلية D وEDC نتيجةً للأخطاء التي تنتج من تغيير بعض البتات (من 1 إلى 0 أو العكس) أثناء انتقال الإطار عبر الوصلة.



الشكل 5-4 سيناريو اكتشاف وتصحيح الأخطاء.

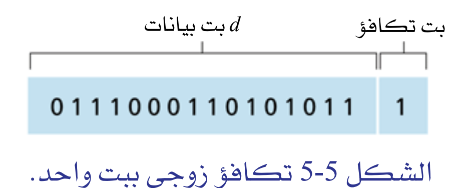
يكمن التحدي الذي يواجهه المُستقبل في تحديد ما إذا كانت البيانات المستلَمة D' هي نفسها البيانات الأصلية المُرسَلة D، علماً بأنه لم يتلقّ سوى D' و'EDC. من المهم ملاحظة التعبير الدقيق لقرار المُستقبل في الشكل 5-4 (نسأل عما إذا كنا قد اكتشفنا وجود خطأ، وليس عما إذا كان هناك خطأ قد حدث فعلاً!). فأساليب اكتشاف الأخطاء وتصحيحها تمكّن المُستقبل أحياناً – ولكن ليس دائماً! – من اكتشاف حدوث أخطاء في البتات. حتى باستعمال بتات لاكتشاف الأخطاء فقد تبقى هناك أخطاء في البتات لا يتسنى اكتشافها (بمعنى أن المُستقبل مع ذلك قد لايدرك أن البيانات المستلَمة فيها بتات خاطئة). ولذا فقد يُسلّم المُستقبل وحدة بيانات غير صحيحة إلى طبقة الشبكة، أو يفوته أن محتويات

أحد حقول ترويسة إطار البيانات فيها خطأ. ومن ثم فإن هدفنا هنا هو اختيار نظام لاكتشاف الأخطاء يقلل إلى درجة مقبولة من احتمال مرور الأخطاء دون ملاحظتها. عموماً تتطلب أساليب اكتشاف الأخطاء وتصحيحها الأكثر تطوراً (أي التي تضمن احتمالاً ضئيلاً لبقاء أخطاء في البتات بدون اكتشاف) أعباءً إضافية أكثر تتمثل في إمكانيات الحساب اللازمة وعدد البتات الإضافية التي ترسل خصيصاً لغرض اكتشاف الأخطاء وتصحيحها.

دعنا الآن نفحص ثلاثة أساليب لاكتشاف الأخطاء في البيانات المُرسَلة: أسلوب فحص التكافؤ (parity check) لتوضيح المبادئ الأساسية وراء اكتشاف الأخطاء وتصحيحها، وأسلوب الفحص بالجمع (checksum) والمستخدم عادةً في طبقة النقل، وأسلوب فحص الفائض الدوري (Cyclic Redundancy Check (CRC)) والمستخدَم عادةً في طبقة ربط البيانات المحققة في بطاقات مواءمة الشبكة.

5-2-1 فحص التكافؤ (Parity Check)

لعل أبسط أشكال اكتشاف الأخطاء هو استخدام بت واحد للتكافؤ. افترض أن وحدة البيانات المطلوب إرسالها في الشكل 5-5 هي D، والتى تتألف من بتات عددها d. في نظامٍ يستخدم فحص التكافؤ الزوجي يضيف المُرسِل بتاً واحداً ويختار قيمتها ببساطة بحيث يكون عدد البتات التي قيمتها 1 ضمن العدد الكلي للبتات $d + 1$ (أي بتات البيانات الأصلية علاوة على البت الإضافي للتكافؤ) عدداً زوجياً. أما في أنظمة فحص التكافؤ الفردي فيتم اختيار قيمة بت التكافؤ بحيث يكون العدد الكلي للبتات التي قيمتها 1 فردياً. يوضح الشكل 5-5 نظام فحص التكافؤ الزوجي حيث يتم تخزين بت التكافؤ الوحيد في حقل مستقل.



الشكل 5-5 تكافؤ زوجي ببت واحد.

باستخدام بت واحد للتكافؤ تكون عملية الفحص في المُستقبِل بسيطةً كذلك، حيث يحتاج المُستقبِل فقط لأن يعد البتات التي قيمتها 1 في الرسالة الكلية التي تم استلامها بطول $(d + 1)$ بت. فإذا وُجد في نظام لفحص التكافؤ الزوجي أن عدد تلك البتات فردي، فإن المُستقبِل يدرك أن خطأً ما قد طرأ في بت واحد على الأقل (وبتحديد أكثر يدرك أن عدداً فردياً من أخطاء البتات قد حدث). لكن ماذا لو حدث عدد زوجي من أخطاء البتات؟ عليك أن تقنع نفسك بأن ذلك سيؤدي إلى خطأ غير مكتشف. إذا كان احتمال حدوث خطأ في البت ضئيلاً وبافتراض أن أخطاء البتات تحدث بشكلٍ مستقل من بت إلى آخر، فإن احتمال حدوث أخطاء في عدة بتات في نفس الإطار يكون ضئيلاً جداً، وفي هذه الحالة قد يكفي بت تكافؤ واحد. ومع ذلك فقد أظهرت القياسات العملية أن أخطاء البتات لا تحدث فقط بشكلٍ مستقل، بل غالباً ما تحدث سويةً على شكل تجمّعات (دفقات) (bursts). عند حدوث أخطاء البتات على شكل تجمعات، يزداد احتمال الأخطاء غير المكتشفة في إطار بيانات محمي ببت تكافؤ واحدٍ ليقارب 50٪ [Spragins 1991]. واضح أننا بحاجة إلى أسلوب أكثر فعالية لاكتشاف الأخطاء. لحسن الحظ هذا الأسلوب موجود، بل ومستخدم عملياً! لكن قبل الانتقال لأساليب اكتشاف الأخطاء المستخدمة في الواقع، دعنا نتناول تعميماً بسيطاً لنظام بت التكافؤ الواحد والذى سيوضح لنا بعض الأمور فيما يتعلق بأساليب تصحيح الأخطاء.

يوضح الشكل 5-6 تعميماً في بعدين لنظام بت التكافؤ الواحد. في هذه الحالة تُقسّم البتات d التي تشكّل قطعة البيانات D المطلوب إرسالها إلى i صف وj عمود، ويتم حساب قيمة بت التكافؤ لكل صف ولكل عمود على حدة بالإضافة إلى حساب بت التكافؤ للقطعة D ككل. تشكّل بتات التكافؤ والتي عددها $i + j + 1$ بتات اكتشاف الأخطاء لإطار طبقة ربط البيانات.

تكافؤ الصفوف

تكافؤ الأعمدة

$d_{1,1}$	...	$d_{1,j}$	$d_{1,j+1}$
$d_{2,1}$	...	$d_{2,j}$	$d_{2,j+1}$
...	...	...	...
$d_{i,1}$	...	$d_{i,j}$	$d_{i,j+1}$
$d_{i+1,1}$	...	$d_{i+1,j}$	$d_{i+1,j+1}$

لا أخطاء

1	0	1	0	1	1
1	1	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

خطأ في بت واحد
يمكن تصحيحه

1	0	1	0	1	1
1	0	1	1	0	0
0	1	1	1	0	1
0	0	1	0	1	0

خطأ تكافؤ

خطأ تكافؤ

الشكل 5-6 تكافؤ زوجي في بُعدين.

لنفترض الآن أن خطأً وقع في بت واحد من بتات البيانات الأصلية والتي عددها d. في هذا النظام ثنائي الأبعاد لفحص التكافؤ سينتج عن ذلك خطأ في تكافؤ كلٍّ من العمود والصف اللذين يحتويان على البت الذي انعكست حالته بسبب الخطأ. وبالتالي يكون بوسع المُستقبِل ليس فقط اكتشاف حدوث خطأ في بت واحد، ولكن أيضاً تحديد موقع ذلك البت بمعلومية موقعي العمود والصف اللذين أظهرا خطأ تكافؤ، ومن ثم تصحيح ذلك الخطأ! يبين الشكل 5-6 مثالاً فيه خطأ في البت الذي موقعه (2، 2) والذى قيمته الأصلية 1، وهو خطأ يمكن اكتشافه، بل وأيضاً تصحيحه عند المُستقبِل. رغم أن مناقشتنا السابقة تركزت على خطأ في بتات البيانات الأصلية التي عددها d، فإنه يمكن أيضاً اكتشاف وتصحيح خطأ

واحد في بتات التكافؤ الإضافية. بوسع نظام فحص التكافؤ ثنائي الأبعاد أيضاً أن يكتشف أي خطأين في إطار البيانات، ولكنه لا يصححها إلا إذا كانا في صفّين مختلفين وعمودين مختلفين. سيتم استكشاف الخواص الأخرى لنظام فحص التكافؤ ثنائي الأبعاد من خلال التمارين الموجودة في نهاية هذا الفصل.

يُطلق على اكتشاف وتصحيح الأخطاء بواسطة المُستقبِل "التصحيح الأمامي للخطأ" (FEC)، ويُستخدم عموماً في تخزين وتشغيل الملفات السمعية كما في حالة الأقراص السمعية المدمجة. ويمكن في مجال الشبكات استخدام أساليب FEC منفردة أو بالاشتراك مع أساليب إعادة الإرسال التلقائي (ARQ) في طبقة ربط البيانات التي تشبه الأساليب التي درسناها في الفصل الثالث. تكمن أهمية أساليب FEC في كونها تقلل من كمية البيانات التي يحتاج المُرسِل لإعادة إرسالها نتيجة حدوث خطأ فيها، كما أنها تسمح بالتصحيح الفوري للأخطاء لدى المُستقبِل وبالتالي تجنب الانتظار لمدة رحلة الذهاب والإياب لتلقّي المُرسِل إشعار استلام سلبي من المُستقبِل، ووصول الإطار المعاد إرساله إليه. لهذه الميزة أهميتها الكبيرة بشكلٍ خاص في التطبيقات الفورية للشبكة [Rubenstein 1998]، ووصلات البيانات ذات تأخيرات الانتقال الطويلة (كوصلات الأقمار الصناعية). من الأبحاث التي تناولت استخدام أساليب FEC في بروتوكولات التحكم في الخطأ: [Biersack 1992; Nonnenmacher 1998؛ Byers 1998؛ Shacham 1990].

5-2-2 أساليب الفحص بالجمع (Checksum)

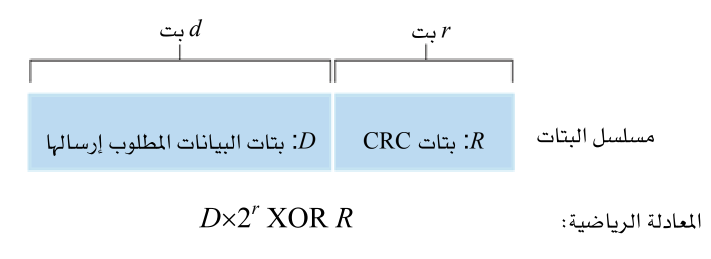
في أساليب الفحص بالجمع تُعتبر رسالة البيانات المكونة من d بت – كما في الشكل 5-5 – بمثابة سلسلة من الأعداد الصحيحة يتألف كلٌّ منها من k بتاً. يتضمن أحد الأساليب البسيطة للفحص جمع تلك الأعداد الصحيحة وإرسال حاصل الجمع الناتج كبتات اكتشاف الأخطاء. وتستخدم هذه الطريقة في الإنترنت حيث تُعدّ بايتات البيانات أعداداً صحيحة يتألف كلٌّ منها من 16 بتاً ويتم جمعها، ثم يُحسب مكمل الواحد (1's complement) لحاصل الجمع ويوضع في ترويسة قطعة البيانات (سنطلق على تلك البتات "المجموع التدقيقي"). كما بيّنا في الجزء 3-3 يقوم

المُستقبِل بإجراء نفس العملية على البيانات التي تم استلامها (بما في ذلك المجموع التدقيقي للبيانات المُرسلِة) لمعرفة ما إذا كانت البتات الناتجة كلها لها القيمة 0. فإذا كان أيٌّ من البتات الناتجة قيمته 1، فإن هذا يدل على وجود خطأ. يناقش طلب التعليقات RFC 1071 خوارزمية الفحص بالجمع في الإنترنت وتطبيقاتها بالتفصيل. يتم حساب المجموع التدقيقي في بروتوكولات TCP وUDP في الإنترنت باستخدام كل الحقول (بما في ذلك حقول البيانات والترويسة). أما في بروتوكول IP فيُحسَب المجموع التدقيقي من بتات ترويسة IP فقط (نظراً لأن كلاً من قطعتي TCP وUDP لهما المجموع التدقيقي الخاص بهما). في البروتوكولات الأخرى (على سبيل المثال بروتوكول XTP [Strayer 1992] يتم حساب مجموع تدقيقي على الترويسة ومجموع تدقيقي آخر على القطعة بأكملها.

تمثّل أساليب الفحص بالجمع عبئاً إضافياً ضئيلاً نسبياً على قطع البيانات المُرسلِة. فعلى سبيل المثال يستخدم المجموع التدقيقي في بروتوكولي TCP وUDP 16 بتاً فقط. غير أنها توفر حماية ضعيفة نسبياً ضد الأخطاء مقارنةً بأسلوب فحص الفائض الدوري (CRC)، والذي سنتناوله لاحقاً والمستخدم غالباً في طبقة ربط البيانات. السؤال الذي يطرح نفسه الآن: لماذا يُستخدم الفحص بالجمع في طبقة النقل بينما يُستخدم أسلوب فحص الفائض الدوري في طبقة ربط البيانات؟ تذكّر أن طبقة النقل تنفّذ عادةً على شكل برامج تمثل جزءاً من نظام التشغيل على المضيف، لذا فمن المهم استخدام طريقة بسيطة وسريعة كالفحص بالجمع. في المقابل ينفّذ اكتشاف الأخطاء في طبقة ربط البيانات في مكونات مادية مخصّصة لذلك في بطاقات مواءمة الشبكة، والتي يمكنها أن تؤدّي العمليات الأكثر تعقيداً ضمن أسلوب فحص الفائض الدوري بسرعة. يعرض [Feldmeier 1995] أساليب برمجية سريعة لتنفيذ العديد من طرق اكتشاف الأخطاء منها الفحص بالجمع الموزون وفحص الفائض الدوري وغيرها.

3-2-5 فحص الفائض الدوري (CRC)

يعتمد أحد أساليب اكتشاف الأخطاء المستخدم بكثرة في شبكات الحاسب اليوم على شفرات فحص الفائض الدوري (CRC)، والتي تُعرف أيضاً بشفرات الدوال متعددة الحدود (polynomials) نظراً لأنه يمكن اعتبار سلسلة البتات المُرسِلة كدالة متعددة الحدود تكون معاملات الحدود فيها هي قيم البتات في السلسلة (0 أو 1) والعمليات على سلسلة البتات كعمليات رياضية على تلك الدوال.



الشكل 5-7 شفرات فحص الفائض الدوري (CRC).

تتلخص طريقة عمل شفرات CRC كالتالي: افترض أن عقدة الإرسال تريد بث قطعة بيانات D طولها d بت إلى عقدة الاستقبال. ينبغي أن يتفق المُرسِل والمُستقبِل بادئٍ ذي بدء على مسلسل بتات مكون من $r+1$ بت ويُعرف بالمولّد(generator)، ونرمز له بالرمز G. سنشترط أن تكون البت في أكبر خانة (أي الخانة في أقصى اليسار) في المولّد G هي 1 دائماً. يوضح الشكل 5-7 الفكرة الرئيسة لعمل أسلوب شفرات CRC. لكل قطعة بيانات D مطلوب إرسالها، يختار المُرسِل قطعة إضافية R طولها r بت، ويلحقها على يمين القطعة D بحيث تقبل القطعة الناتجة بطول $d + r$ بت (باعتبارها عدداً ثنائياً) القسمة بالضبط (أي بدون باقٍ) على المولّد G باستعمال حساب الباقي الثنائي (modulo-2 arithmetic). وبهذا تكون عملية التدقيق بحثاً عن الأخطاء بسيطة، حيث يَقسِم المُستقبِل القطعة التي استلمها بطول $d + r$ بت على المولّد G. فإذا وجد أن باقي القسمة ليس صفراً، يعرف

المُستقبِل إن خطأً ما قد طرأ على البيانات أثناء انتقالها من المُرسِل؛ وإلا فإنه يفترض أن البيانات التي وصلته صحيحة.

تتم كل عمليات شفرات CRC باستخدام حساب modulo-2 بدون حَمل (carry) من خانة إلى الخانة التي تليها في عمليات الجمع، أو استلاف (borrow) إلى خانة من الخانة التي تليها في عمليات الطرح. هذا يعني أن الجمع والطرح هنا عمليتان متكافئتان، وكلاهما يكافئ العملية المنطقية "أو - الحصرية"(XOR) عند إجرائها على كل زوج من البتات على حدة. فعلى سبيل المثال:

$$1011 \text{ XOR } 0101 = 1110$$

$$1001 \text{ XOR } 1101 = 0100$$

وبالمثل نحصل أيضاً على:

$$1011 - 0101 = 1110$$

$$1001 - 1101 = 0100$$

يتم الضرب والقسمة كما في الحساب الثنائي (base-2 arithmetic)، فيما عدا أن أي عمليات جمع أو طرح مطلوبة تُجرى بدون حمل أو استلاف كما ذكرنا أعلاه. كما في الحساب الثنائي العادي يؤدي ضرب عدد في 2^k إلى إزاحة مسلسل بتات العدد إلى اليسار k خانة. وهكذا فبمعلومية D وR فإن العملية:

$$D \times 2^r \text{ XOR } R$$

تنتج مسلسل بتات بطول $d + r$ والمبين في الشكل 5-7. سنستخدم هذا التمثيل الجبري لمسلسل البتات بطول $d + r$ في الشكل 5-7 في مناقشتنا التالية.

لنلتفت الآن للسؤال الجوهري: كيف يحسب المُرسِل العدد R؟ تذكّر أننا نريد إيجاد R بحيث يكون هناك عدد n يحقق العلاقة:

$$D \times 2^r \text{ XOR } R = n \times G$$

أي أننا نريد اختيار R بحيث إن G تقسم $(D\times2^r \text{ XOR } R)$ بدون باقٍ. إذا قمنا بعملية XOR لـ R على الطرفين (أي أضفنا R بحساب modulo-2 بدون حمل من خانة إلى خانة) فإننا نحصل على:

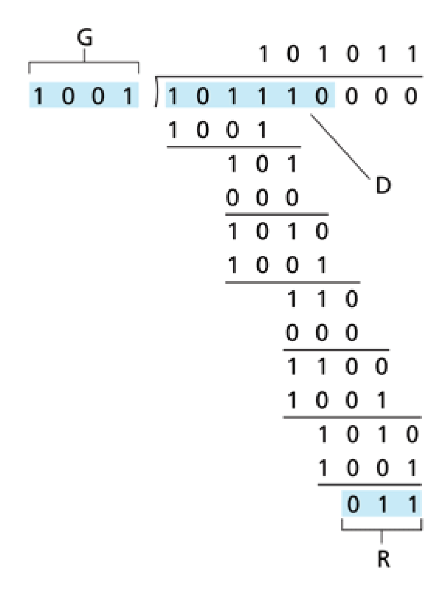
$$D\times2^r = n\times G \text{ XOR } R$$

تخبرنا هذه المعادلة بأننا إذا قسمنا $D\times2^r$ على G فإن الباقي يكون R بالضبط. بمعنى آخر يمكننا حساب R كالتالي:

$$R = \text{remainder}\left(\frac{D\times2^r}{G}\right)$$

يوضح الشكل 5-8 هذه العملية الحسابية للحالة التي فيها: $D = 101110$، $d = 6$، $G = 1001$ وبالتالي $r = 3$. عندئذٍ تكون البتات التسعة التي يتم إرسالها هي 101110 011. عليك التأكد من أن:

$$D\times2^r = 101011\times G \text{ XOR } R$$



الشكل 5-8 مثال لحساب شفرة فحص الفائض الدوري (CRC).

تم وضع مواصفات معيارية دولية لمولدات G بمقاسات: 8، 12، 16، 32 بتاً. تُستخدم شفرة CRC بمقاس 8 بتات لحماية ترويسة تضم 5 بايتات في وحدات بيانات شبكات ATM (أو ما سنطلق عليه "خلايا"). يُستخدم CRC-32 المعياري بمقاس 32 بت والمنفّذ في عددٍ من بروتوكولات IEEE لطبقة ربط البيانات المولّد:

$$G_{\text{CRC-32}} = 100000100110000010001110110110111$$

بوسع كلٍّ من شفرات CRC المعيارية اكتشاف تجمع أخطاء (burst) طوله أقل من $r + 1$ بت (أي سيتم اكتشاف كل الأخطاء في r بت متتالية أو أقل). علاوة على ذلك ـ عند تحقق الفرضيات المناسبة ـ سيتم اكتشاف تجمع أخطاء بطول أكبر من $r + 1$ بت بإحتمال 0.5^r - 1. وأيضاً يمكن لكل شفرات CRC المعيارية اكتشاف أي عدد فردي من أخطاء البتات. راجع [Williams 1993] لمناقشةٍ لتنفيذ العمليات المتعلقة بشفرات CRC. إن نظرية شفرات CRC والشفرات الأخرى الأكثر فعالية في اكتشاف أخطاء البيانات وتصحيحها تقع خارج نطاق هذا الكتاب، ويمكنك الاطلاع على [Schwartz 1980] والذى يتضمن مقدمة ممتازة عن هذا الموضوع.

3-5 بروتوكولات الوصول المتعدد

في مقدمة هذا الفصل ذكرنا أن هناك نوعين من وصلات الشبكة: الوصلات من نقطة إلى نقطة (point-to-point links) ووصلات الإذاعة (broadcast links). تشمل الوصلة من نقطة إلى نقطة مُرسِلاً واحداً على أحد طرفي الوصلة ومُستقبِلاً واحداً على طرفها الآخر. تم تصميم العديد من البروتوكولات للعمل على الوصلات من نقطة إلى نقطة، ومنها بروتوكول "نقطة إلى نقطة" (PPP) وبروتوكول "التحكم عالي المستوى في وصلة البيانات" (High-Level Data Link Control (HDLC)) واللذان سنغطيهما لاحقاً في هذا الفصل. أما النوع الثاني من الوصلات فيتضمن عدة عقد للإرسال والاستقبال موصّلة بقناة إذاعة وحيدة ومشتركة بينهم (سنستخدم مصطلح "إذاعة" هنا لأنه عندما تبث إحدى العقد إطار بيانات تتم إذاعته على قناة الوصلة وتتلقى كل عقدة من العقد الأخرى نسخة منه).

من أمثلة تقنيات وصلات ربط البيانات بالإذاعة شبكات البيانات المحلية (LANs) من نوع إيثرنت (Ethernet) ومن النوع اللاسلكي (wireless). في هذا الجزء سنأخذ خطوة إلى الوراء مبتعدين قليلاً عن البروتوكولات المحددة لطبقة ربط البيانات لندرس أولاً مشكلة ذات أهمية جوهرية لعمل طبقة الربط، وهي كيفية تنسيق وصول عدة عقد للإرسال والاستقبال لقناة إذاعة وحيدة مشتركة – أي مشكلة الوصول المتعدد. غالباً ما تستخدم قنوات الإذاعة في شبكات البيانات المحلية – وهي شبكات تقع جغرافياً في مبنى واحد (أو داخل شركة أو حرم جامعي) – ولذا سنتناول كيفية استخدام قنوات الوصول المتعدد في شبكات البيانات المحلية في نهاية هذا الجزء.

كلنا على دراية بفكرة الإذاعة – فالتلفزيون يستخدمها منذ نشأته. لكن الاتصال في التلفزيون التقليدي أحادي الاتجاه، حيث تقوم عقدة ثابتة واحدة بالبث للعديد من العقد التي تستقبل ذلك البث؛ بينما يمكن لكل عقدة على قناة إذاعة ضمن شبكة حاسب الإرسال والاستقبال. لعل المثال الأكثر ملاءمةً لحالة قناة الإذاعة هو تجمع الناس في حفل بقاعة كبيرة يتحدثون ويستمعون لبعضهم البعض (حيث يوفر الهواء الوسط المادي للإذاعة). مثال آخر جيد ومألوف لدى العديد من القرّاء هو قاعة الدرس حيث يستخدم المعلّم والطلاب نفس وسط الإذاعة الوحيد بنفس الطريقة. من المشاكل الجوهرية في كلا المثالين تقرير من الذي يتكلم (أي يبث على القناة) ومتى. لقد طوّر البشر مجموعة متقنة من البروتوكولات للاشتراك في استخدم قناة إذاعة، مثل:

"أعط كل شخص فرصةً للحديث."

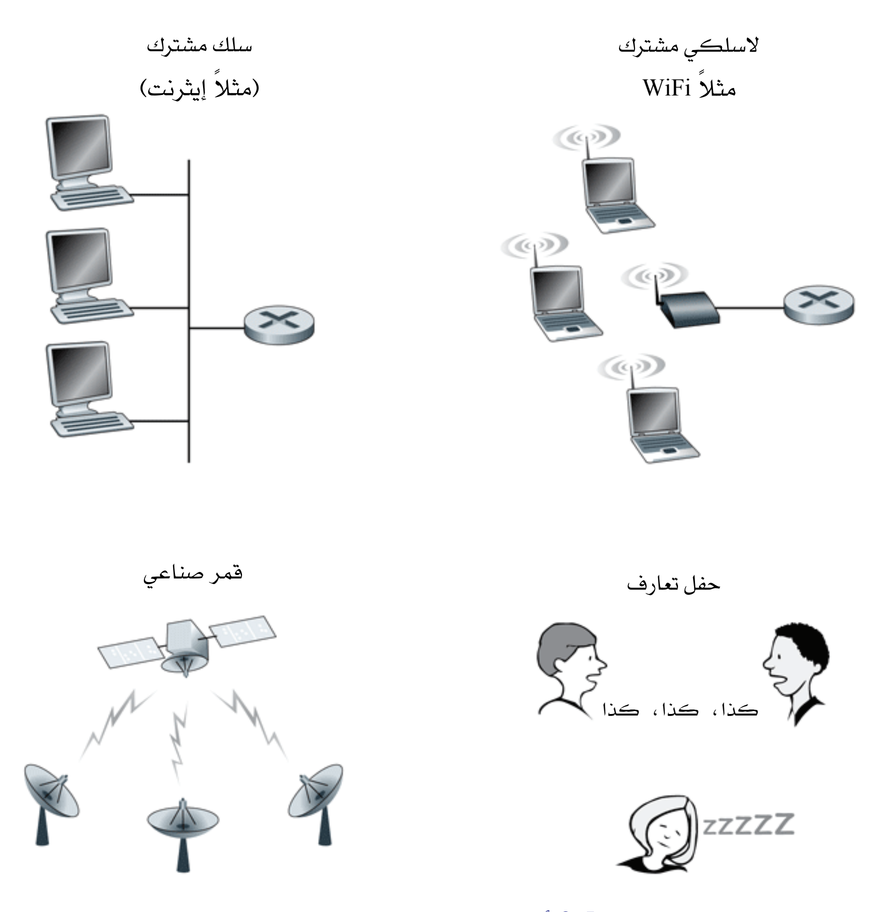
"لا تتحدث إلا إذا تحدث شخص إليك."

"لا تحتكر المحادثة."

"ارفع يدك إذا كان لديك سؤال."

"لا تقاطع شخصاً يتحدث."

"لا تنعس بينما شخص يتحدّث."



الشكل 5-9 أمثلة متنوعة لقنوات الوصول المتعدد.

بالمثل فإن لشبكات الحاسب ما يعرف ببروتوكولات الوصول المتعدد والتي تستخدمها العُقد في تنظيم إرسالها على قناة الإذاعة المشتركة. كما يبين الشكل 5-9 نحتاج لبروتوكولات الوصول المتعدد في العديد من الشبكات بما في ذلك شبكات البيانات المحلية – السلكية منها واللاسلكية – وشبكات الأقمار الصناعية. رغم أنه من الناحية الفنية توصّل كل عقدة بقناة الإذاعة من خلال بطاقة مواءمة، إلا أننا للتبسيط سنعتبر في هذا الجزء أن العقدة هي نفسها أداة الإرسال والاستقبال. بوسع المئات بل الآلاف من العقد الاتصال مباشرةً عبر قناة إذاعة.

نظراً لأنه بوسع كل العقد إرسال إطارات بيانات، يمكن أن تقوم أكثر من عقدتين بإرسال إطارات في نفس الوقت. عندما يحدث ذلك تتلقى كل العقد على قناة الإذاعة المشتركة عدة إطارات في نفس الوقت مما يؤدي إلى تصادم الإطارات المُرسَلة عند كل العقد المُستقبِلة، وفي حالة حدوث ذلك لا تستطيع أيٌّ من العقد المُستقبِلة فهم أيٍ من الإطارات التي أُرسِلت. ويرجع ذلك إلى تداخل إشارات الإطارات المصطدمة بشكلٍ معقد، ومن ثم تعتبر كل الإطارات المشتركة في الاصطدام مفقودة، وبالتالي لا تتحقق أي فائدة من قناة الإذاعة أثناء فترة الاصطدام. واضح أنه إذا كان هناك العديد من العقد التي تريد إرسال الإطارات بكثرة، فإن نسبةً كبيرةً من عمليات الإرسال ستؤدي إلى اصطدامات، وسيضيع جزء كبير من الحيز الترددي (سعة الإرسال) لقناة الإذاعة سُدىً.

لكي نضمن قيام قناة الإذاعة المشتركة بعمل مفيد عند تفعيل عقد متعددة من الضروري تنسيق عمليات الإرسال بين تلك العقد بطريقة ما. إن هذا التنسيق هو مسؤولية بروتوكول الوصول المتعدد. خلال الأعوام الثلاثين الماضية كُتبت آلاف الأبحاث والمئات من أطروحات الدكتوراه حول هذا الموضوع، ويتضمن [Rom 1990] مسحاً شاملاً لهذا الجهد. وعلاوة على ذلك فالبحث في مجال بروتوكولات الوصول المتعدد مازال نشطاً بسبب ظهور أنواع جديدة من الوصلات باستمرار، وبخاصة الوصلات اللاسلكية الجديدة.

على مرّ السنين استُخدمت العشرات من بروتوكولات الوصول المتعدد ضمن الأنواع المختلفة من تقنيات طبقة ربط البيانات. ومع ذلك يمكننا تصنيف أي بروتوكول للوصول المتعدد تقريباً ضمن واحد من الأصناف الثلاثة التالية: بروتوكولات تقسيم القناة، وبروتوكولات الوصول العشوائي للقناة، وبروتوكولات التناوب على القناة. سنغطّي هذه الأصناف الثلاثة من بروتوكولات الوصول المتعدد في الأجزاء الثلاثة التالية.

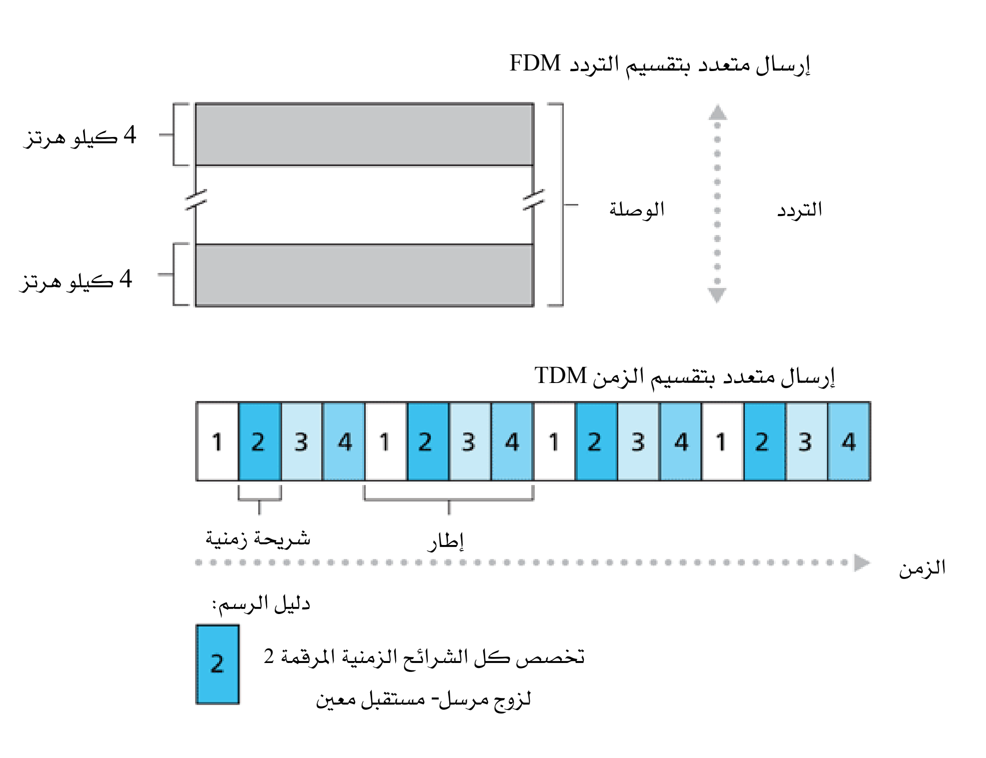
دعنا نختم هذا الاستعراض العام بالملاحظة التالية: في الحالة المثالية عند استخدام قناة إذاعة مشتركة لها سعة إرسال R بت/ثانية يجدر ببروتوكول الوصول المتعدد تحقيق الخصائص المرغوبة التالية:

1. عندما تكون عقدة واحدة فقط لديها بيانات للإرسال، يتم توفير طاقة إنتاجية قدرها R بت/ثانية لتلك العقدة.
2. عندما تكون هناك M عقدة لديها بيانات للإرسال يتم توفير طاقة إنتاجية قدرها R/M بت/ثانية لكلٍّ منها، ولا يلزم بالضرورة تحقيق معدّل إرسال آني قدره R/M بصفة دائمة، ولكن يكفي توفير معدّل إرسال متوسط قدره R/M لكل عقدة على مدى فترة زمنية يتم تحديدها بشكلٍ مناسب.
3. أن يكون البروتوكول غير مركزي؛ بمعنى ألاّ يعتمد في تنفيذه على وجود عقد رئيسة (سيادية) قد تتعرض للتعطل ومن ثم تؤدي إلى تعطل النظام بالكامل.
4. أن يكون البروتوكول بسيطاً بحيث يمكن تنفيذه بكلفة قليلة.

5-3-1 بروتوكولات تقسيم القناة

تذكّر من مناقشتنا السابقة في الجزء 1-3 أنه يمكن استخدام تقنية الإرسال المتعدد بتقسيم الزمن (Time-Division Multiplexing (TDM)) أو بتقسيم التردد (Frequency-Division Multiplexing (FDM)) لإشراك كل العقد في الحيز الترددي لقناة إذاعة مشتركة. كمثال افترض أن القناة تدعم N عقدة وأن معدل الإرسال المسموح به على القناة هو R بت/ثانية. تقوم تقنية TDM بتقسيم الوقت إلى إطارات زمنية (time frames) ثم تقسّم كل إطار بدوره إلى N شريحة زمنية، وتُخصّص شريحةً زمنيةً لكل واحدة من العقد التي عددها N. ينبغي عدم الخلط بين إطار TDM الزمني ووحدة تبادل البيانات في طبقة ربط البيانات والتي يطلق عليها أيضاً اسم إطار. لكي نقلل من احتمال حدوث هذا الخلط سنطلق في هذا الجزء على وحدة تبادل البيانات في طبقة ربط البيانات اسم رزمة (packet). عندما يكون لدى عقدة رزمة تريد إرسالها فإنها تقوم بإرسال تلك الرزمة أثناء الشريحة الزمنية

المخصصة لها في إطار TDM الدوّار. عادةً ما يتم اختيار مدة الشريحة الزمنية بحيث يمكن إرسال رزمة واحدة أثناء كل شريحة. يبين الشكل 5-10 مثالاً مبسطاً لتقنية TDM بأربع عقد. وبتطبيق ذلك على مثال الحفل المذكور آنفاً، يسمح هذا البروتوكول لكلٍّ من مرتادي الحفل بالحديث لمدة محددة من الوقت ويتوقف بعدها ليتيح الفرصة لشخص آخر للحديث لنفس الفترة، وهكذا. عند الانتهاء من إتاحة الفرصة لكل شخص ليقول ما لديه تُعاد الكَرّة من جديد.



الشكل 5-10 مثال لتقنيات TDM وFDM بأربع عقد.

تعتبر تقنية TDM مرغوبة من حيث إنها تمنع الاصطدام وتعتبر عادلة جداً، فهي تخصص لكل عقدة معدّل إرسال قدره R/N بت/ثانية خلال وقت كل إطار. ومع ذلك فهي تعاني من عيبين رئيسين، أولهما أن معدل الإرسال المتوسط المتاح لعقدة لن يتجاوز R/N بت/ثانية حتى ولو كانت هي العقدة الوحيدة التي لديها رزم

للإرسال. أما العيب الثاني فهو أنه يتعين على كل عقدة دائماً انتظار دورها في طابور الإرسال ـ مرةً أخرى حتى ولو كانت هي العقدة الوحيدة التي لديها رزم للإرسال. تخيّل أن أحد الحضور في الحفل هو الشخص الوحيد الذي لديه ما يقوله، وتخيّل الحالة الأندر التي يكون فيها كل الحضور يريدون سماع ما يقوله ذلك الشخص. من الواضح أن تقنية TDM ستكون اختياراً سيئاً كبروتوكول وصول متعدد لذلك الحفل.

بينما تقسّم تقنية TDM وقت استغلال قناة الإذاعة المشتركة بين العقد على الوصلة، تقوم تقنية FDM بتقسيم الحيز الترددي للقناة (بسعة إرسال R بت/ثانية) إلى نطاقات تردد مختلفة (لكلٍّ منها سعة إرسال قدرها R/N بت/ثانية) وتخصّص كل نطاق لاستخدام عقدة من العقد التي عددها N. وبهذا يكوّن FDM عدد N من القنوات الأصغر لكلٍّ منها سعة إرسال قدرها R/N بت/ثانية من القناة الأكبر ذات سعة الإرسال R بت/ثانية. تشترك تقنية FDM مع تقنية TDM في المزايا التي ذكرناها أعلاه، فهي تتفادى حدوث الاصطدام وتقسّم الحيز الترددي بإنصاف بين العقد. وبالمثل فإنها تشترك معها أيضاً في العيب الرئيس، ألا وهو أن سعة الإرسال المتاحة للعقدة ستكون محدودة بـ R/N بت/ثانية حتى ولو كانت هي العقدة الوحيدة التي يتوافر لديها رزم تودّ إرسالها.

يُعد بروتوكول الوصول المتعدد بتقسيم الشفرات (CDMA) بروتوكولاً ثالثاً لتقسيم القناة. بينما تخصّص تقنية TDM وتقنية FDM شرائح زمنية ونطاقات تردد على التوالي للعقد، يخصّص بروتوكول CDMA شفرةً مختلفةً لكل عقدة، حيث تستخدم كل عقدة شفرتها الفريدة لتشفير بتات البيانات التي ترسلها. إذا تم اختيار الشفرات بعناية تتوافر لشبكات CDMA الخاصية الرائعة التي تسمح للعقد المختلفة بالإرسال في نفس الوقت، ومع ذلك يستطيع كل مُستقبِل استلام بتات البيانات المشفّرة والمُرسِلة إليه بشكلٍ صحيح (بافتراض أن المُستقبِل يعرف شفرة المُرسِل) على الرغم من التداخل بسبب الإرسال من العقد الأخرى. لقد استُخدمت تقنية CDMA في الأنظمة العسكرية لبعض الوقت (بسبب خاصية مقاومة التشويش التي تتمتع بها) ولها الآن استخدامات مدنية على نطاق واسع، خصوصاً في شبكات

الهاتف الخلوي. نظراً لأن استعمال تقنية CDMA وثيق الصلة جداً بقنوات اللاسلكي، فسنؤجل مناقشتنا للتفاصيل الفنية لتقنية CDMA إلى الفصل السادس. أما الآن فيكفي أن نعرف أن الشفرات في CDMA – كما هو الحال مع شرائح الوقت في TDM ونطاقات التردد في FDM – يمكن تخصيصها لمستخدمي القناة المشتركة للوصول المتعدد.

5-3-2 بروتوكولات الوصول العشوائي

المجموعة الثانية من البروتوكولات العامة للوصول المتعدد هي بروتوكولات الوصول العشوائي. في بروتوكول للوصول العشوائي تقوم العقدة بالإرسال دائماً بمعدّل الإرسال الأقصى للقناة، أي R بت/ثانية. عند حدوث اصطدام تقوم كل عقدة اشتركت في الاصطدام بإعادة إرسال إطارها (أو بمعنى آخر رزمتها) مراراً وتكراراً إلى أن يتمكن الإطار من المرور بدون اصطدام. غير أنه عندما تواجه عقدة اصطداماً فإنها لا تعيد إرسال الإطار بعد ذلك مباشرةً بالضرورة، ولكنها بدلاً من ذلك تنتظر لمدة تأخير عشوائية قبل إعادة إرسال الإطار. تختار كل عقدة اشتركت في اصطدام تأخيرات عشوائية مستقلة. ولأن التأخيرات العشوائية يتم اختيارها بشكلٍ مستقل فمن المحتمل أن إحدى العقد ستختار تأخيراً يقل عن تأخيرات عقد الاصطدام الأخرى بما فيه الكفاية بحيث تستطيع أن تدفع بإطارها إلى القناة بدون اصطدام.

هناك العشرات بل المئات من بروتوكولات الوصول العشوائي الموجودة على الساحة [Rom 1990؛ Bertsekas 1991]. في هذا الجزء سنتناول عدداً من بروتوكولات الوصول العشوائي الأكثر استعمالاً: بروتوكولات ألوها (ALOHA) [Abramson 1970; Abramson 1985] وبروتوكولات الوصول المتعدد بالإنصات للناقل ((Carrier Sense Multiple Access (CSMA)) [Kleinrock 1975b]. سنغطّي لاحقاً ـ في الجزء 5-5 ـ تفاصيل الإيثرنت [Metcalfe 1976] وهي بروتوكول مشهور ومستخدم بكثرة من نوع CSMA.

بروتوكول ألوها الشرائحي

دعنا نبدأ دراستنا لبروتوكولات الوصول العشوائي بواحد من أبسط تلك البروتوكولات، ألا وهو بروتوكول ألوها الشرائحي (Slotted ALOHA). في وصفنا لهذا البروتوكول سنفترض الآتي:

- كل الإطارات تتكون من L بت بالضبط.
- يُقسّم الوقت إلى شرائح مدة كلٍّ منها L/R ثانية (أي أن الشريحة الزمنية تكفي لإرسال إطار واحد فقط).
- تبدأ العقد ببث إطاراتها في بدايات الشرائح فقط.
- هناك تزامن بين العقد بحيث تعرف كل عقدة وقت بدأ الشرائح الزمنية.
- إذا اصطدم إطاران أو أكثر فإن كل العقد تكتشف حدوث الاصطدام قبل أن تنتهي الشريحة الزمنية التي حدث فيها.

افترض أن p تمثل احتمالاً، أي أن قيمتها تتراوح ما بين 0 و1. إن تنفيذ بروتوكول ألوها الشرائحي في كل عقدة هو عملية بسيطة تتلخص في الخطوات التالية:

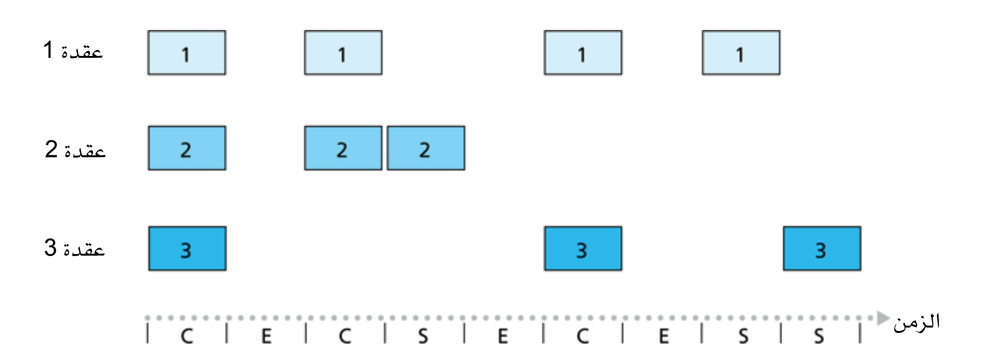
- عندما يكون لدى العقدة إطار جديد تريد إرساله فإنها تنتظر حتى بداية الشريحة التالية، وترسل الإطار بكامله أثناء تلك الشريحة.
- إذا لم يحدث اصطدام تكون العقدة قد أرسلت إطارها بنجاح، ومن ثم لا تحتاج لإعادة إرسال الإطار (بل يمكن أن تجهز العقدة إطاراً جديداً لإرساله إن وُجد).
- أما في حالة وجود اصطدام، فتكتشف العقدة الاصطدام قبل نهاية الشريحة الزمنية، وتعيد محاولة إرسال إطارها في كل شريحة تالية باحتمال p إلى أن يتم إرسال الإطار بدون اصطدام.

نعني بـ "إعادة الإرسال باحتمال p" أنّ العقدة عملياً ترمي قطعة عملة معدنية منحازة؛ حيث يناظر الحدث "صورة" "إعادة إرسال" (ويتكرر باحتمال p)، ويناظر الحدث "كتابة" "انتظر حتى تنتهي هذه الشريحة ثم ارمِ قطعة العملة مرة ثانية في الشريحة التالية" (ويتكرر باحتمال $(1 - p)$). تقوم كل العقد التي اشتركت في الاصطدام برمي قطع العملة لديها بشكلٍ مستقل.

يتضح أن لبروتوكول ألوها الشرائحي العديد من المزايا، فبخلاف بروتوكولات تقسيم القناة يسمح البروتوكول للعقدة بالإرسال بشكلٍ مستمر بمعدل الإرسال الكامل R بت/ثانية عندما تكون تلك العقدة هي العقدة الوحيدة النشطة على القناة (توصف العقدة بأنها نشطة إذا كان لديها إطارات للإرسال). كما أن بروتوكول ألوها الشرائحي بروتوكول بسيط للغاية، ويمتاز أيضاً بأنه غير مركزي بشكلٍ كبير حيث إن كل عقدة تكتشف الاصطدام وتقرر متى تعيد الإرسال بشكلٍ مستقل. ومع ذلك فإن البروتوكول يتطلب تزامن الشرائح لدى العقد. سنناقش بعد قليل نوعية غير شرائحية من ذلك البروتوكول، وكذلك بروتوكولات CSMA، والتي لا يتطلب أيٌّ منها تحقيق أي تزامن، مما يجعلها غير مركزية تماماً.

يعمل بروتوكول ألوها الشرائحي بشكلٍ جيد عندما تكون هناك عقدة نشطة واحدة، لكن ما مدى كفاءته في وجود عدة عقد نشطة؟ هناك عاملان قد يؤثران سلباً على الكفاءة:

- أولاً: كما هو موضح في الشكل 5-11، عند وجود عدة عقد نشطة ستعاني نسبة معينة من الشرائح الزمنية من حدوث اصطدامات أثناءها، ومن ثم ستُهدر تلك الشرائح.
- ثانياً: هناك نسبة أخرى من الشرائح الزمنية ستبقى غير مستغلّة (فارغة) إثر حدوث اصطدام نتيجةً لامتناع كل العقد النشطة عن الإرسال لاتباعها سياسة الاحتمالات. الشرائح الوحيدة التي لن تضيع هباءً ستكون تلك التي تقوم فيها عقدة واحدة فقط بالإرسال (وعندئذ يطلق عليها شريحة ناجحة). سنعرّف كفاءة أي بروتوكول شرائحي للوصول المتعدد بأنها نسبة الشرائح الناجحة على المدى البعيد عند وجود عدد كبير من العقد النشطة لدى كل منها عدد كبير من الإطارات التي تريد إرسالها. لاحظ أنه في غياب أي شكل من أشكال التحكم في الوصول، وقيام كل عقدة بإعادة الإرسال فوراً عقب كل اصطدام، ستكون الكفاءة صفراً. واضح أن كفاءة بروتوكول ألوها الشرائحي تزيد شيئاً ما عن الصفر، ولكن بكَم؟



الشكل 5-11 تصطدم العقد 1 و2 و3 في الشريحة الأولى. تنجح العقدة 2 أخيراً في الشريحة الرابعة، والعقدة 1 في الشريحة الثامنة، والعقدة 3 في الشريحة التاسعة.

نمضي الآن في اشتقاق تعبير رياضي للكفاءة القصوى لبروتوكول ألوها الشرائحي. لتبسيط هذا الاشتقاق دعنا نعدّل البروتوكول قليلاً بأن نفترض أن كل عقدة تحاول إرسال إطارٍ في كل شريحة زمنية باحتمال p (بمعنى أننا نفترض أن كل عقدة لديها دائماً إطار للإرسال، وأن العقدة ترسل الإطارات باحتمال p دائماً سواءً الإطار الجديد أو الذي عانى من اصطدام). افترض أن لدينا N عقدة، عندئذ يكون احتمال أن شريحة بعينها هي شريحة ناجحة هو احتمال قيام إحدى العقد بالإرسال بينما تمتنع بقية العقد الـ $(N - 1)$ عن الإرسال. احتمال قيام عقدة بالإرسال هو p، واحتمال عدم قيام كل العقد الباقية بالإرسال هو $(1 - p)^{N-1}$ وعليه يكون احتمال نجاح عقدة بعينها هو $p(1 - p)^{N-1}$. ونظراً لأننا لدينا N عقدة، فإن احتمال نجاح أي عقدة في الإرسال هو $Np(1 - p)^{N-1}$.

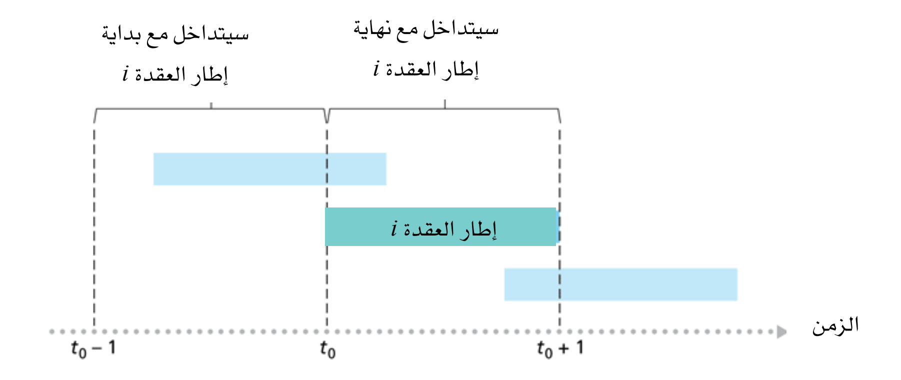
وبالتالي فعند وجود عقد نشطة تكون كفاءة بروتوكول ألوها الشرائحي $Np(1- p)^{N-1}$. للحصول على الكفاءة القصوى لـ N عقدة نشطة، علينا إيجاد القيمة p^* التي تحقق أقصى قيمة لذلك التعبير (راجع تمارين هذا الفصل للاطلاع على استعراض عام لهذا الاشتقاق). وللحصول على الكفاءة القصوى لعدد كبير من

العقد النشطة، نأخذ نهاية $Np^*(1-p^*)^{N-1}$ بينما تقترب N من ما لانهاية (مرةً أخرى راجع التمارين في نهاية الفصل). بعد القيام بهذه الحسابات سنجد أن الكفاءة القصوى للنظام تساوي تقريبا $1/e = 0.37$. أي أنه في وجود عدد كبير من العقد لديها العديد من الإطارات لإرسالها، فإنه (في أحسن الأحوال) تُستخدم 37 بالمائة من الشرائح الزمنية فقط للقيام بعمل مفيد. وعليه فإن نسبة الإرسال الفعّالة للقناة ليست R بت/ثانية ولكن فقط $0.37\ R$ بت/ثانية. يبين تحليل مماثل أن 37 بالمائة من الشرائح تذهب فارغة و26 بالمائة منها تعاني من اصطدامات. تخيّل خيبة أمل مشرف الشبكة الذي اشترى نظام ألوها الشرائحي للعمل على قناة بسعة إرسال 100 ميجابت/ثانية متوقعاً أن يكون بوسعه استعمال الشبكة لإرسال البيانات بين عدد كبير من المستخدمين بمعدل إرسال كلي قدره مثلاً 80 ميجابت/ثانية! رغم أن القناة قادرة على إرسال الإطار بمعدل الإرسال الكامل للقناة (100 ميجابت/ثانية)، فإنه على المدى البعيد ستكون الطاقة الإنتاجية الناجحة لتلك القناة أقل من 37 ميجابت/ثانية.

بروتوكول ألوها

يتطلّب بروتوكول ألوها الشرائحي من كل العقد أن تُزامِن إرسالها بحيث يبدأ مع بداية الشريحة الزمنية. غير أن بروتوكول ألوها الأصلي [Abramson 1970] كان في الواقع بروتوكولاً غير شرائحي وغير مركزي تماماً. ففي ذلك البروتوكول عندما يصل إطار لأول مرة (بتمرير قطعة بيانات في عقدة الإرسال من طبقة الشبكة إلى طبقة ربط البيانات) تُرسِل العقدة الإطارَ كله فوراً عبر قناة الإذاعة المشتركة. وإذا واجه إطارٌ مُرسَلٌ اصطداماً مع واحد أو أكثر من الإطارات المُرسَلة، فإن العقدة تعيد إرسال ذلك الإطار فوراً (بمجرد الانتهاء من إرسال الإطار المصطدم)، وذلك بالاحتمال p. وإلا فإن العقدة تنتظر (تبقى عاطلة) لفترة إرسال إطار، وبعدها ترسل الإطار بالإحتمال p أو تنتظر لفترة إرسال إطار آخر باحتمال $(1-p)$.

لتعيين الكفاءة القصوى لبروتوكول ألوها الأصلي سنركّز على عقدة بعينها. سنفترض نفس فرضيات التحليل السابق لبروتوكول ألوها الشرائحي، ونفترض أن وقت إرسال الإطار يمثل وحدة الزمن. في أي وقت يكون احتمال قيام العقدة بإرسال إطار هو p. افترض أن العقدة i تبدأ في إرسال هذا الإطار في الوقت t_0. كما هو مبيّن في الشكل 5-12 لكي يتم إرسال هذا الإطار بنجاح ينبغي ألا تبدأ أي عقدة أخرى إرسالها خلال الفترة $[t_0-1, t_0]$، لأن مثل هذا الإرسال يتداخل مع بداية إرسال إطار العقدة i. احتمال أن كل العقد الأخرى لا تبدأ إرسالها خلال تلك الفترة هو $(1-p)^{N-1}$. بالمثل لا ينبغي أن تقوم عقدة أخرى بالإرسال بينما العقدة i ترسل، حيث إن ذلك يتداخل مع الجزء الأخير من إرسال إطار العقدة i. احتمال أن كل العقد الأخرى لا تبدأ الإرسال في تلك الفترة هو أيضاً $(1-p)^{N-1}$. وعليه فإن احتمال أن تتمكن عقدة بعينها من القيام بإرسال ناجح هو $p(1-p)^{2(N-1)}$. بأخذ النهايات كما في حالة بروتوكول ألوها الشريحي، نجد أن الكفاءة القصوى لبروتوكول ألوها الأصلي هي $(1/2e)$ فقط (أي بالضبط نصف القيمة لبروتوكول ألوها الشرائحي). هذا إذن هو الثمن الذي ندفعه مقابل استخدام بروتوكول ألوها غير المركزي تماماً.



الشكل 5-12 تداخل عمليات الإرسال في بروتوكول ألوها.

تاريخ حالة (Case History)

نورم أبرامسون وشبكة ألوهانت:

نورم أبرامسون هو مهندس يحمل شهادة الدكتوراه، وقد كان لديه هواية التزلج على الماء واهتمام بتحويل رزم البيانات. قاده هذا الاهتمام إلى جامعة هاواي في عام 1969، ونظراً لأن هاواي تضم مجموعة من الجزر الجبلية فقد كان تركيب وتشغيل الشبكات الأرضية أمراً صعباً. عندما لم يكن أبرامسون يتزلج على الماء، كان يفكر كيف يصمّم شبكة تقوم بتحويل رزم البيانات على موجات الراديو. تضمنت الشبكة التي صمّمها مضيفاً مركزياً واحداً وعدّة عقد ثانوية مبعثرة على جزر هاواي، وكانت تستخدم قناتين لكلٍ منهما نطاق ترددي مختلف. استُخدِمت قناة الوصلة الهابطة (downlink) لإذاعة الرزم من المضيف المركزي إلى المضيفات الثانوية، بينما استخدمت قناة الوصلة الصاعدة (uplink) لإرسال الرزم من المضيفات الثانوية إلى المضيف المركزي. علاوةً على إرسال رزم المعلومات كان المضيف المركزي يرسل أيضاً على قناة الوصلة الهابطة إشعار استلام لكل رزمة يتم استلامها بنجاح من المضيفات الثانوية.

ونظراً لأن المضيفات الثانوية ترسل الرزم بشكلٍ غير مركزي كانت الاصطدامات تحدث حتماً على قناة الوصلة الصاعدة. قادت تلك الملاحظة أبرامسون لابتكار بروتوكول ألوها الأصلي كما وصفناه من قبل في هذا الفصل. في عام 1970 وبتمويل مستمر من الوكالة الأمريكية لمشاريع البحوث المتقدمة (ARPA)، قام أبرامسون بتوصيل شبكة ألوهانت (ALOHAnet) إلى شبكة أربانت (ARPAnet). تكمن أهمية عمل أبرامسون ليس فقط في كونه أول شبكة راديو من نوعها لتحويل الرزم، ولكن أيضاً لأنه كان مصدر إلهام لبوب ميتكالف (Bob Metcalfe). فبعد سنوات قليلة عدّل ميتكالف بروتوكول ألوها ليبتكر بروتوكول الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام CSMA/CD وشبكة الإيثرنت المحلية.

الوصول المتعدد بالإنصات للناقل (CSMA)

في بروتوكول ألوها – بكلا نوعيه الشرائحي والأصلي – تتخذ كل عقدة قرارها بالإرسال أو عدمه بشكلٍ مستقل عن نشاط العقد الأخرى المشتركة معها في نفس قناة الإذاعة، وبالتحديد لا تكترث العقدة إذا صادفت عقدةً أخرى تقوم بالإرسال في الوقت ذاته التي تشرع هي فيه ببدء الإرسال، كما أنها لا توقف إرسالها إذا ما بدأت عقدة أخرى بالتداخل مع ما ترسله. في مثال الحفل الذي ذكرناه آنفاً تشبه بروتوكولات ألوها تماماً مرتاد الحفل الفظ الذي يواصل

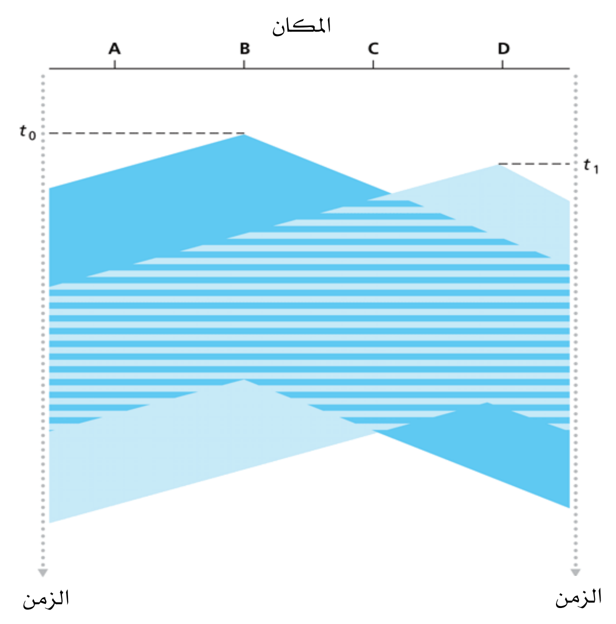
دردشة سواء كان الآخرون يتكلمون أم لا. إننا كبشر لدينا بروتوكولات تجعلنا نتصرّف ليس فقط بلطف ولكن أيضاً بحيث نقلل الوقت الذي "تصطدم" فيه محادثاتنا مع الآخرين، ومن ثم زيادة كمية البيانات التي نتبادلها من خلال تلك المحادثات. وبشكلٍ محدّد هناك قاعدتان ذهبيتان للمحادثة البشرية الدمثة:

1. استمع قبل الكلام، فإذا كان هناك شخص آخر يتكلم فانتظر حتى ينتهي. في عالم الشبكات يُعرف هذا الأسلوب بالإنصات للناقل، حيث تنصت العقدة إلى القناة قبل الشروع في الإرسال. إذا حدث وكانت هناك عقدة أخرى تبث إطاراً حالياً على القناة فإن العقدة التي تُنصت تنتظر (تتراجع) لفترة عشوائية من الوقت وبعد ذلك تنصت للقناة مرةً أخرى. إذا وجدت العقدة أن القناة خالية فإنها تبدأ إرسال إطارها، وإلا فإنها تنتظر لفترة عشوائية أخرى وتكرّر العملية.
2. إذا بدأ شخص آخر الكلام في نفس الوقت توقّف أنت عن الكلام. يُعرف هذا في عالم الشبكات باكتشاف الاصطدام، حيث تنصت العقدة المُرسِلة للقناة أثناء قيامها بالإرسال وإذا اكتشفت أن عقدة أخرى ترسل إطاراً يتداخل مع إطارها الذي ترسله، فإنها تتوقّف عن الإرسال وتستخدم بعض قواعد البروتوكول لتحديد متى يمكنها إعادة محاولة الإرسال مرةً أخرى.

تم تضمين هاتين القاعدتين في عائلة بروتوكولات الوصول المتعدد بأسلوب الإنصات للناقل والوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام (CSMA/CD) [Kleinrock 1975b; Metcalfe 1976; Lam 1980; Rom 1990]. تم اقتراح العديد من أنواع بروتوكولات CSMA وCSMA/CD، ويمكنك الرجوع لتلك المراجع للاطلاع على تفاصيل تلك البروتوكولات. سندرس نظام CSMA/CD المستخدم في شبكات الإيثرنت بالتفصيل في الجزء 5-5. أما هنا فسنلقي الضوء على بعض الخصائص الهامة والأساسية لبروتوكولات CSMA و CSMA/CD.

لعل السؤال الذي سيتبادر إلى ذهنك للوهلة الأولى عن بروتوكول CSMA هو: لماذا تحدث الاصطدامات أساساً إذا كانت كل العقد تنصت للناقل؟ فكل عقدة ستمتنع عن الإرسال عندما تشعر بأن عقدة أخرى ترسل. لعل أفضل طريقة لتوضيح الإجابة عن هذا التساؤل هي استخدام مخططات المكان والزمن [Molle 1987].

يبين الشكل 5-13 مخطط المكان والزمن لأربع عقد (A, B, C, D) موصلة على ناقل إذاعة خطي (linear broadcast bus). يبين المحور الأفقي موقع كل عقدة على الناقل بينما يمثل المحور العمودي الزمن.

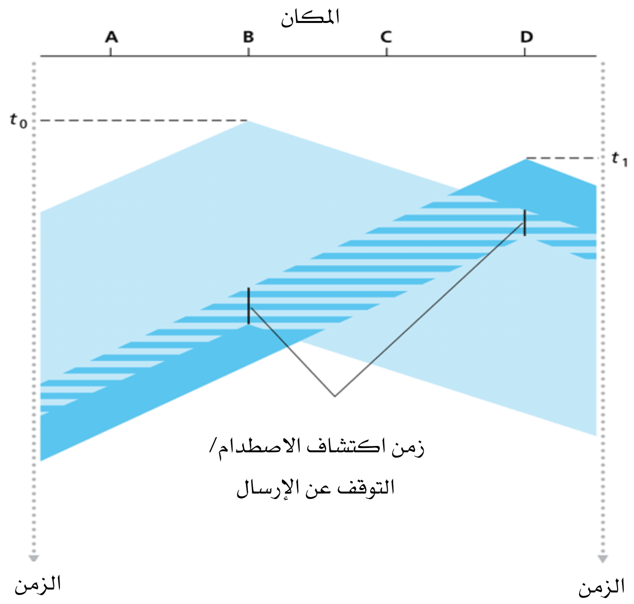


الشكل 5-13 مخطط المكان والزمن لعقدتي بروتوكول CSMA مع اصطدام للإرسال.

عند النقطة t_0 من الزمن تحس العقدة B أن القناة خالية، حيث لا توجد عقد أخرى تقوم بالإرسال حالياً. وعليه تبدأ العقدة B بالإرسال، فتنتقل البتات التي ترسلها في كلا الإتجاهين على طول وسط الإذاعة المشترك. إن انتقال بتات العقدة B إلى أسفل مع زيادة الوقت في الشكل 5-13 يبين أن تلك البتات تحتاج لفترة محددة من الوقت (ليست صفراً) لانتقال البتات على طول وسط الإذاعة المشترك (رغم انتقالها بسرعة كبيرة تقارب سرعة الضوء) . عند اللحظة t_1 — حيث $(t_1 > t_0)$ — يتوافر لدى العقدة D إطارٌ للإرسال. رغم أن العقدة B تقوم فعلاً بالإرسال في اللحظة t_1، إلا أن البتات التي ترسلها B لم تصل بعد إلى D، ومن ثم تحس D أن القناة خالية عند t_1. تبعاً لبروتوكول CSMA تبدأ D بإرسال إطارها. بعد مرور فترة قصيرة من الوقت، يأخذ إرسال B في التداخل مع إرسال D. يتبين من الشكل 5-13 أن

تأخير الانتقال من طرف إلى طرف عبر قناة الإذاعة المشتركة يلعب دوراً حاسماً في تحديد أداء هذا النظام. فكلما كان هذا التأخير أطول ازداد احتمال عدم تمكن عقدة تنصت للناقل من الإحساس بإرسال بدأته عقدة أخرى على الشبكة.

في الشكل 5-13 لا تقوم العقد باكتشاف الاصطدام، فكل من العقدتين B وD تواصل إرسال إطاراتها كاملةً رغم حدوث اصطدام. عندما يتوافر لعقدة إمكانية اكتشاف الاصطدام، سوف توقف إرسالها بمجرد اكتشافها وقوع الاصطدام. يبين الشكل 5-14 نفس السيناريو الموضح في الشكل 5-13 فيما عدا أن العقدتين توقفان إرسالهما بعد فترة وجيزة من اكتشاف الاصطدام. واضح أن إضافة إمكانية اكتشاف الاصطدام لبروتوكول الوصول المتعدد بالإنصات للناقل سيحسّن أداء البروتوكول، وذلك بمنع إرسال الإطار عديم الفائدة بالكامل (أي الإطار الذي فسد بسبب التداخل مع إطارٍ مُرسَلٍ من عقدة أخرى). بروتوكول الإيثرنت الذي سندرسه في الجزء 5-5 هو بروتوكول من هذا النوع (أي الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام).



الشكل 5-14 بروتوكول الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام.

5-3-3 بروتوكولات التناوب على القناة

تذكّر أنه من الخواص المرغوبة في بروتوكول الوصول المتعدد: (1) عند وجود عقدة واحدة نشطة، تتوافر لتلك العقدة طاقة إنتاجية R بت/ثانية، و(2) عند وجود M عقدة نشطة، تتوافر لكل عقدة نشطة طاقة إنتاجية مقدارها R/M بت/ثانية تقريباً. يلاحظ أن بروتوكولات ألوها وCSMA تتحقق فيها الخاصية الأولى، لكن لا تتحقق فيها الخاصية الثانية. لقد حفّز هذا الأمر الباحثين لتطوير طائفة أخرى من البروتوكولات يطلق عليها بروتوكولات التناوب على القناة. كما هو الحال مع بروتوكولات الوصول العشوائي، هناك العشرات من بروتوكولات التناوب على القناة، ولكل واحدٍ منها العديد من النوعيات المختلفة. سنتناول هنا اثنين من أهم تلك البروتوكولات. يتطلّب الأول، وهو بروتوكول الاستفتاء (polling)، تعيين إحدى العقد كعقدة رئيسة (master node). تقوم العقدة الرئيسة باستطلاع وضع كلٍّ من العقد الأخرى بشكلٍ دوري لمعرفة ما إذا كان لديها ما تريد إرساله. وبالتحديد ترسل العقدة الرئيسة أولاً رسالة إلى العقدة 1 مفادها أنها (أي العقدة 1) يمكنها إرسال عدة إطارات كحدٍ أقصى يتم تعيينه في الرسالة. بعد انتهاء العقدة 1 من إرسال إطاراتها، تخبر العقدة الرئيسة عقدة 2 أنه بوسعها (أي العقدة 2) إرسال العدد الأقصى من الإطارات. يمكن للعقدة الرئيسة تحديد ما إذا كانت عقدة مُرسِلة قد انتهت من إرسال إطاراتها بملاحظة غياب الإشارة على القناة. تستمر العملية بهذه الطريقة، حيث تستطلع العقدة الرئيسة كل عقدة من العقد بطريقة دورية.

يتخلص بروتوكول الاستفتاء من الاصطدامات ومن ترك الشرائح الزمنية فارغة ـ وهي عيوب تعاني منها بروتوكولات الوصول العشوائي ـ وبالتالي يمكنه تحقيق كفاءة أعلى بكثير. غير أنه يعاني أيضاً من عدة عيوب. العيب الأول: هو أنه يتضمن تأخيراً جديداً هو تأخير الاستطلاع (أي الوقت اللازم لإخبار عقدة أنها يمكنها أن ترسِل). فمثلاً إذا كانت هناك عقدة واحدة نشطة، فإنها سترسل البيانات بمعدل إرسال أقل من R بت/ثانية، حيث إنه على العقدة الرئيسة استطلاع

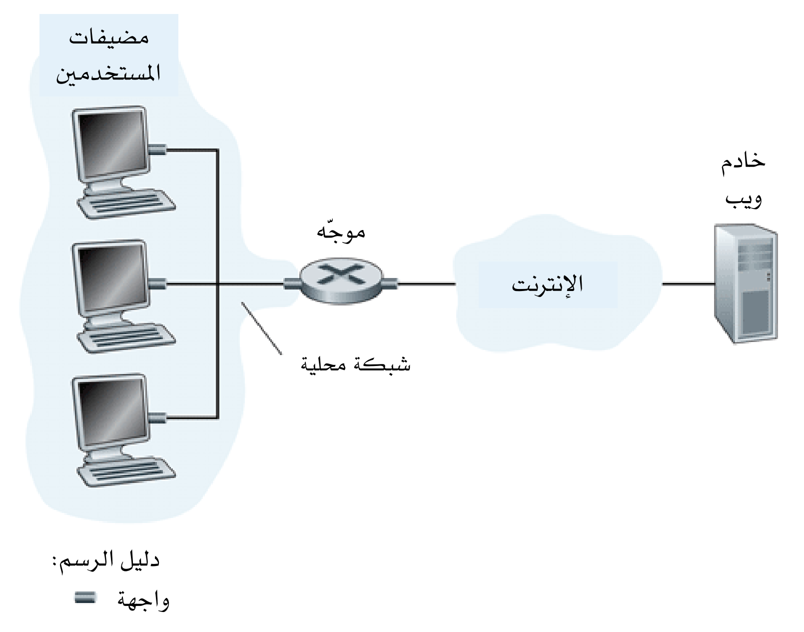
وضع العقد الخاملة تباعاً كلما انتهت العقدة النشطة من إرسال العدد الأقصى المحدد لها من الإطارات. أما العيب الثاني: وهو الأشد خطورة، فيكمن في العقدة الرئيسة؛ لأنه في حال تعطلها لا يمكن تشغيل القناة.

البروتوكول الثاني للتناوب على القناة هو بروتوكول تمرير العلامة ("التوكن") (token-passing). في هذا البروتوكول لا توجد عقدة رئيسة، وإنما يتم تبادل إطار خاص صغير يُعرف بالعلامة بين العقد بترتيب ثابت. فمثلاً قد ترسل العقدة 1 العلامة دائماً إلى العقدة 2، والعقدة 2 قد ترسلها دائماً إلى العقدة 3، والعقدة N قد ترسلها دائماً إلى العقدة 1. عندما تسلّم العلامة لعقدةٍ ما فإن العقدة تحتفظ بها فقط إذا كان لديها بعض الإطارات تريد إرسالها، وإلا فإنها ترسل العلامة مباشرة إلى العقدة التالية. إذا كان لدى عقدة إطارات للإرسال عندما تستلم العلامة، فإنها ترسل العدد الأقصى المسموح به من الإطارات ثم تمرر العلامة إلى العقدة التالية. يعتبر أسلوب تمرير العلامة غير مركزيٍ وذا كفاءة عاليةً، ولكن له مشاكله أيضاً. على سبيل المثال، قد يؤدي تعطل عقدة واحدة إلى تعطل القناة بأكملها. كما أنه إذا أهملت عقدة ما بشكلٍ عرَضي تمرير العلامة فسيحتاج الأمر إلى إجراءٍ للتعافي من هذا الخطأ واستئناف عملية تمرير العلامة. تم تطوير عدة بروتوكولات لتمرير العلامة على مدار سنين عديدة، وكل واحد منها كان عليه التصدي لتلك المشاكل وغيرها من القضايا المتعلقة. سنذكر اثنين من تلك البروتوكولات في الجزء التالي: بروتوكول FDDI وبروتوكول IEEE 802.5.

5-3-4 شبكات البيانات المحلية (LANs)

تُستخدَم بروتوكولات الوصول المتعدد مع العديد من الأنواع المختلفة لقنوات الإذاعة المشتركة، حيث تستخدم مع القنوات اللاسلكية وقنوات الأقمار الصناعية والتى تقوم فيها العقد بالإرسال على نفس النطاق الترددي، وتُستخدم حالياً للوصول للإنترنت عن طريق قناة الوصلة الصاعدة للكبل (انظر الجزء 1-2)، كما تُستخدم بكثرة على شبكات البيانات المحلية (LANs).

تذكّر أن شبكة البيانات المحلية LAN هي شبكة حاسب مركّزة في منطقة جغرافية كبناية أو حرم جامعي. عندما يدخل مستخدم على الإنترنت من جامعة أو مقر شركة، يكون الوصول غالباً عن طريق شبكة بيانات محلية. وبالتحديد يتم الوصول من المضيف إلى الشبكة المحلية إلى الموجّه إلى الإنترنت كما هو مبين في الشكل 5-15. جدير بالذكر أن معدل الإرسال R لمعظم شبكات البيانات المحلية عالٍ جداً. حتى في أوائل الثمانينيات كانت الشبكات المحلية التى تعمل بسرعات 10 ميجابت/ثانية منتشرة. واليوم تتوافر الشبكات المحلية بمعدلات إرسال قدرها 100 ميجابت/ثانية و1 جيجابت/ثانية و10 جيجابت/ثانية.



الشكل 5-15 وصول المضيفات إلى خادم الويب على الإنترنت عن طريق شبكة بيانات محلية. تتألف قناة الإذاعة المشتركة بين المضيفات والموجّه من وصلة واحدة.

في الثمانينيات وأوائل التسعينيات ظهر صنفان من تقنيات شبكات البيانات المحلية وانتشرا في أماكن العمل. شمل الصنف الأول شبكات الإيثرنت المحلية المعروفة بشبكات IEEE 802.3 [IEEE 802.3 2007]، وهي مصممّة على أساس الوصول العشوائي. أما الصنف الثاني من شبكات البيانات المحلية فقد تضمّن تقنيات تمرير العلامة (token-passing)، ومن بينها حلقة العلامة (token ring) (والمعروفة كذلك بـ IEEE 802.5 [IEEE 802.5 2007]، وواجهة البيانات الموزعة عبر الألياف الضوئية (FDDI) [Jain 1994]. نظراً لأننا سنتناول تقنيات الإيثرنت بشيء من التفصيل في الجزء 5-5، فسوف نركّز مناقشتنا هنا على شبكات البيانات المحلية بتمرير العلامة. وستكون مناقشتنا لتقنيات تمرير العلامة قصيرة عن قصد؛ لأن المنافسة المستمرة من قِبَل الإيثرنت قد جعلت تلك التقنيات شبه منقرضة الآن تقريباً. ومع ذلك، ولكي نقدم بعض الأمثلة لتقنية تمرير العلامة ونعطي منظوراً تاريخياً مبسطاً من المفيد ذكر نبذة مختصرة عن شبكات حلقة العلامة.

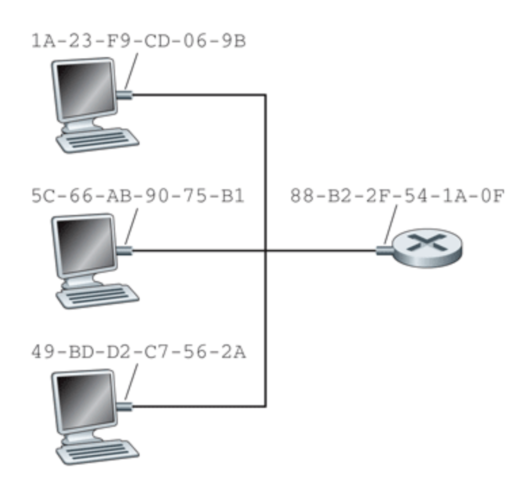
في شبكة بيانات محلية من نوع حلقة العلامة تُوصّل عُقد الشبكة (افترض أن عددها N وأنها تضم مضيفات وموجّهات) على شكل حلقة باستخدام وصلات مباشرة. تحدد طبوغرافية حلقة العلامة الترتيب المتّبع لتمرير العلامة. عندما تحصل عقدة على العلامة وترسل إطاراً، ينتقل الإطار حول الحلقة بأكملها، وبذلك تنشأ قناة إذاعة افتراضية. تقرأ العقدة المقصودة (الوجهة) الإطار من الوسط المادي لطبقة ربط البيانات أثناء مرور الإطار بها. تتحمّل العقدة التي ترسل الإطار مسؤولية إزالة الإطار من الحلقة. بهذا الأسلوب صُمّمت واجهة البيانات الموزعة عبر الألياف الضوئية (FDDI) لشبكات البيانات المحلية التي تمتد جغرافياً على مساحات أكبر، بما في ذلك شبكات المنطقة الحَضَرية (Metropolitan Area Networks (MANs)). في شبكات البيانات المحلية الممتدة جغرافياً عبر عدّة كيلومترات يقلل من كفاءة الشبكة ترك الإطار ينتقل مرةً أخرى إلى العقدة المُرسِلة بعد عبوره عقدة الوجهة. لذا ففي شبكات FDDI تقوم عقدة الوجهة نفسها بإزالة الإطار من الحلقة (وعليه فإن شبكة FDDI ليست بالضبط قناة إذاعة بالمعنى الحرفي، حيث إن كل عقدة لا تستلم كل إطار يتم إرساله).

5-4 العنونة في طبقة ربط البيانات

للعقد – أي المضيفات والموجّهات ـ عناوين في طبقة ربط البيانات. الآن قد تجد في هذا مفاجأة، بعد أن عرفت في الفصل الرابع أن العقد لها أيضاً عناوين بطبقة الشبكة. وقد تتساءل الآن لماذا نحتاج لأن يكون لدينا عناوين في كلٍّ من طبقة الشبكة وطبقة ربط البيانات؟ بالإضافة إلى وصف قواعد ووظائف عناوين طبقة ربط البيانات، نأمل أن نتمكن في هذا الجزء من توضيح كيف أن وجود طبقتين من العنونة هو أمر مفيد، بل وفي حقيقة الأمر لا غنى عنه. سنغطي أيضاً بروتوكول تحويل العناوين ((Address Resolution Protocol (ARP)، والذي يوفّر آلية لترجمة عناوين طبقة الشبكة IP إلى عناوين طبقة ربط البيانات.

5-4-1 عناوين طبقة ربط البيانات

في الحقيقة، ليست العقدة ـ أي المضيف أو الموجّه ـ هي التي لها عنوان طبقة ربط البيانات ولكن موائم الشبكة بالعقدة هو الذي له عنوان طبقة ربط البيانات. يوضح هذا المفهوم الشكل 5-16. يُطلق على عنوان طبقة ربط البيانات أسماء مختلفة، كعنوان شبكة البيانات المحلية (LAN address)، والعنوان المادي (physical address)، أو عنوان طبقة ربط البيانات (عنوان الماك) (MAC address). ونظراً لأن التعبير الأخير يبدو أكثر تلك التعابير شهرة، فسوف نشير لعناوين طبقة ربط البيانات ابتداءً من الآن بعناوين الماك. في معظم شبكات البيانات المحلية (بما في ذلك الإيثرنت وشبكة البيانات المحلية اللاسلكية 802.11)، يتكون عنوان الماك من 6 بايتات، ومن ثم يسمح بـ 2^{48} عنوان ماك مختلف. كما هو مبين في الشكل 5-16، يُعبّر عن تلك العناوين المؤلفة من 6 بايتات عادةً بصيغة أعداد ستة عشرية (0-9, A-F)، حيث يمثل كل بايت من بايتات العنوان بزوج من الأعداد الستة عشرية. رغم أن عناوين الماك صمّمت لتكون ثابتة، فمن الممكن الآن تغيير عنوان الماك لموائم الشبكة عن طريق البرامج. على كل حال فإننا طوال هذا الجزء سنفترض أن عنوان الماك لموائم الشبكة هو عنوان ثابت.



الشكل 5-16 لكل موائم موصل بشبكة البيانات المحلية عنوان ماك فريد.

من الخواص الشائقة لعناوين الماك عدم وجود موائمين لهما نفس العنوان. قد يبدو ذلك مفاجأة لك. السؤال الآن: إذا كانت تلك الموائمات يتم إنتاجها في العديد من البلدان بواسطة العديد من الشركات، فكيف لشركة تنتج الموائمات في تايوان أن تتأكّد من أنها تستعمل عناوين مختلفة عن تلك التي تستخدمها شركة أخرى تنتج الموائمات في بلجيكا؟ الجواب على ذلك هو أن منظمة IEEE تدير فضاء عناوين الماك. وبتحديدٍ أكثر عندما تريد شركة صناعة موائمات، فإنها تشتري حيزاً من فضاء عناوين الماك يضم 2^{24} عنواناً مقابل أجر معين. تخصّص IEEE للشركة 2^{24} عنواناً بتثبيت الـ 24 بتاً الأولى من بتات عنوان الماك، وتترك للشركة الحرية لتكوين عناوين ماك فريدة تناظر التباديل المختلفة لقيم البتات الـ 24 الأخيرة من عنوان الماك لكل موائم.

عنوان الماك لموائم له تركيب مسطّح (flat) (في مقابل التركيب الهرمي hierarchical) ولا يتغيّر أينما ذهب الموائم. فحاسبٌ نقّال مزوّد ببطاقة إيثرنت يكون له نفس عنوان الماك دائماً أينما ذهب ذلك الحاسب. ومساعد شخصي رقمي (PDA) بموائم لاسلكي 802.11 يكون له نفس عنوان الماك دائماً أينما ذهب ذلك الـ PDA. تذكّر أنه على النقيض من ذلك يكون لعناوين طبقة الشبكة (IP addresses)

تركيب هرمي (أي أن العنوان يتكون من جزءٍ خاصٍ بالشبكة وجزءٍ خاصٍ بالمضيف)، وعليه فإن عنوان IP لمضيف ينبغي تغييره عند انتقال المضيف (أي عند تغيير الشبكة الموصّل بها). يشبه عنوان الماك الخاص بالموائم رقم الضمان الاجتماعي للشخص، والذي له أيضاً تركيب مسطح ولا يتغيّر أينما ذهب الشخص. أما عنوان IP فيماثل العنوان البريدي للشخص، والذي له تركيب هرمي ويلزم تغييره عندما ينتقل الشخص. تماماً كما يجد الشخص من المفيد أن يكون له عنوان بريدي ورقم ضمان اجتماعي، فمن المفيد للعقدة أن يكون لها عنوان بطبقة الشبكة (IP address) وعنوان ماك.

كما ذكرنا في بداية هذا الجزء، عندما يريد موائمٌ إرسال إطار إلى موائم وجهة، يقوم موائم المُرسِل بوضع عنوان الماك لموائم الوجهة بالإطار، وبعد ذلك يرسل الإطار إلى شبكة البيانات المحلية. إذا كانت الشبكة من نوع شبكات الإذاعة (كشبكة 802.11 والعديد غيرها من شبكات الإيثرنت المحلية) يتم استلام الإطار ومعالجته بواسطة كل الموائمات الأخرى الموصّلة على الشبكة المحلية. وبالتحديد يقوم كل موائم يتلقّى الإطار بالتأكد مما إذا كان عنوان الماك للوجهة والموجود في الإطار يوافق عنوان الماك الخاص بالموائم. إذا كان الأمر كذلك، ينتزع الموائم قطعة البيانات المرفقة بالإطار، ويدفع بها لأعلى عبر رصة البروتوكولات على عقدته الأم. إذا لم يحدث تطابق بين العنوانين، يهمل الموائم الإطار ولا يمرّر وحدة بيانات طبقة الشبكة لأعلى عبر رصة البروتوكولات. وهكذا فإن عقدة الوجهة فقط هي التي سيتم مقاطعتها عند استلام الإطار.

ومع ذلك ففي بعض الأحيان يريد موائم المُرسِل من كل الموائمات الأخرى على الشبكة المحلية أن تستلم وتعالج الإطار الذي سيرسله. في هذه الحالة يضع موائم الإرسال عنوان ماك مخصص لوظيفة الإذاعة (broadcast address) في حقل عنوان الوجهة بالإطار. في الشبكات المحلية التي تستخدم عناوين طولها 6 بايتات (كشبكات إيثرنت وشبكات تمرير العلامة يكون العنوان المخصص لإذاعة الإطار هو سلسلة من 48 بتاً قيمة كل منها 1 (أي FF-FF-FF-FF-FF-FF بالترقيم الست عشري).

المبادئ في الواقع العملي (Principles in Practice)

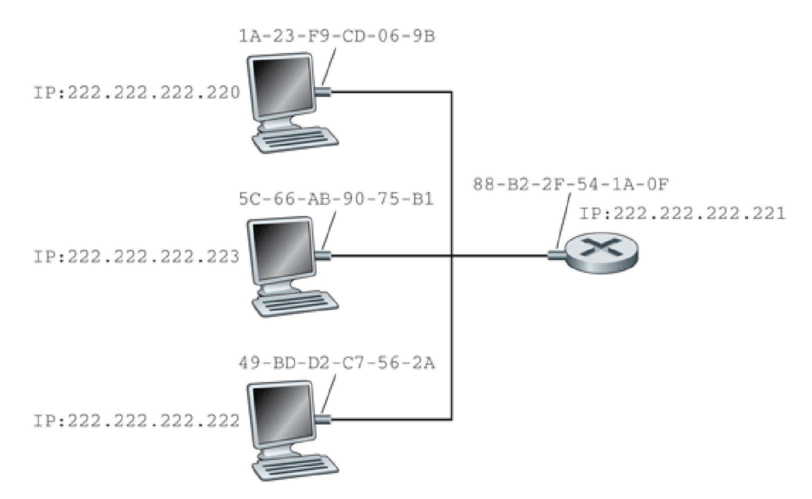
الحفاظ على استقلال الطبقات

هناك العديد من الأسباب لإعطاء العقد على الشبكة عناوين مادية (ماك) بالإضافة إلى عناوين طبقة الشبكة. أولاً: صممت شبكات البيانات المحلية (LANs) لبروتوكولات مختلفة لطبقة الشبكة وليست فقط لبروتوكولات الإنترنت. إذا خُصّصت للموائمات عناوين IP بدلاً من عناوين الماك "المحايدة"، فلن يكون من السهل عليها التعامل مع بروتوكولات طبقة الشبكة الأخرى (على سبيل المثال IPX أو DECnet). ثانياً: إذا كان على الموائمات استخدام عناوين طبقة الشبكة بدلاً من عناوين ماك، فسيتعين تخزين عنوان طبقة الشبكة في ذاكرة القراءة والكتابة (RAM) للموائم والتي ستحتاج بالتالي إلى إعادة تهيئتها في كل مرة يُنقل فيها الموائم إلى مكان جديد (أو يعاد تشغيله بعد إطفائه). هناك خيار آخر هو عدم استخدام أي عناوين للموائمات وجعل كل موائم يُمرّر البيانات (عادةً وحدة بيانات طبقة الشبكة) الموجودة في كل إطار يستلمه إلى أعلى عبر رصة البروتوكول. يمكن أن تقوم طبقة الشبكة في هذه الحالة بالتأكد من توافق عنوان طبقة الشبكة. من مشاكل هذا الخيار أنه ستتم مقاطعة المضيف عند وصول كل إطار يرسل على الشبكة المحلية، بما في ذلك الإطارات الموجّهة لعقد أخرى على نفس وصلة الإذاعة بالشبكة المحلية. الخلاصة هي أنه لكي تكون الطبقات وحدات بناء مستقلة بشكلٍ كبير في البنية المعمارية للشبكة، تحتاج الطبقات المختلفة لنظام عنونة خاص بها. لقد رأينا حتى الآن ثلاثة أنواع من العناوين: أسماء المضيفات في طبقة التطبيقات، وعناوين IP في طبقة الشبكة، وعناوين الماك في طبقة ربط البيانات.

5-4-2 بروتوكول تحويل العناوين (ARP)

نظراً لوجود عناوين لطبقة الشبكة (مثلاً عناوين IP الخاصة بالإنترنت) وعناوين لطبقة ربط البيانات (أي عناوين الماك)، هناك حاجة للتحويل بينهما. في حالة الإنترنت يضطلع بهذه المهمة بروتوكول تحويل العناوين (Address Resolution Protocol (ARP)) [RFC 826].

لفهم الحاجة إلى مثل هذا البروتوكول خذ في الاعتبار الشبكة المبيّنة في الشكل 5-17. في هذا المثال البسيط لكل عقدة عنوان IP واحد، ولكل موائم عنوان ماك واحد. كالمعتاد تبيّن عناوين IP بالصيغة العشرية المنقوطة، بينما تبيّن عناوين الماك بالترقيم الست عشري. افترض الآن أن العقدة بعنوان IP 222.222.222.220 تريد إرسال قطعة بيانات IP إلى العقدة 222.222.222.222 (على سبيل المثال قد تكون عقدة الوجهة 222.222.222.222 خادم ويب، وقد تكون عقدة الإرسال 222.222.222.220 قد حددت عنوان IP لخادم الويب بواسطة بروتوكول DNS). في هذا المثال تقع كل من عقدتي المصدر والوجهة في نفس شبكة البيانات المحلية حسب مفهوم العنونة الذي تناولناه في الجزء 4-4-2. لإرسال حزمة بيانات يجب على عقدة المصدر أن تعطي موائمها ليس فقط وحدة بيانات IP، ولكن أيضاً عنوان الماك لعقدة الوجهة 222.222.222.222. بتوفر وحدة بيانات IP ومعلومية عنوان الماك، يقوم موائم عقدة الإرسال بتكوين إطار طبقة ربط البيانات يحتوي على عنوان الماك لعقدة الوجهة ثم يرسل الإطار إلى الشبكة المحلية.



الشكل 5-17 لكل عقدة على شبكة البيانات المحلية عنوان IP، ولكل موائم عقدة عنوان ماك.

السؤال المهم الذي نتناوله في هذا الجزء هو: كيف تحدد عقدة الإرسال عنوان الماك لعقدة الوجهة التي لها عنوان IP 222.222.222.222؟ تقوم بذلك باستخدام بروتوكول تحويل العناوين (ARP)، حيث تأخذ وحدة بروتوكول ARP الموجودة على عقدة الإرسال أي عنوان IP على نفس الشبكة المحلية كمُدخل وتُرجع عنوان الماك المقابل. في المثال الذي نحن بصدده تزوّد عقدة الإرسال 222.222.222.220 وحدة ARP عليها بعنوان IP 222.222.222.222، فتُرجع لها وحدة بروتوكول ARP عنوان الماك المقابل 49-BD-D2-C7-56-2A.

وهكذا نرى أن بروتوكول ARP يحوّل عنوان IP إلى عنوان ماك. في الكثير من الجوانب يشبه ذلك بروتوكول خدمة الدليل لأسماء النطاقات (DNS) لتحويل أسماء المضيفات إلى عناوين IP، والذي سبق أن درسناه في الجزء 2-5. غير أن هناك فرقاً جوهرياً بين تحويل العناوين في الحالتين، فبينما يحوّل بروتوكول DNS أسماء المضيفات الموجودة في أي مكان على الإنترنت، يحوّل بروتوكول ARP عناوين IP إلى عناوين الماك فقط للعقد على نفس الشبكة الفرعية (subnet). إذا حاولت عقدة في كاليفورنيا استخدام بروتوكول ARP لتحويل عنوان IP لعقدة في ميسيسيبى، فإن بروتوكول ARP يُرجع تنبيهاً بحدوث خطأ.

الآن بعد أن وضّحنا الدور الذي يقوم به بروتوكول ARP لتحويل العناوين، دعنا ننظر كيف يؤدي البروتوكول هذا الدور. تحتوي كل عقدة (مضيف أو موجّه) في ذاكرة القراءة والكتابة بها على جدول لتحويل عناوين IP إلى عناوين الماك المقابلة. يبين الشكل 5-18 كيف يمكن أن يبدو جدول ARP على العقدة 222.222.222.220. يتضمن الجدول كذلك فترة العمر ((Time-To-Live (TTL)) لكل مُدخل (صف) والتي تبين مدة الاحتفاظ بالصف في الجدول. لاحظ أن الجدول لا يحتوي بالضرورة على مُدخل لكل عقدة على الشبكة الفرعية، فبعض العقد ربما تكون قد انتهت صلاحية المُدخلات الخاصة بها، والبعض الآخر ربما لم يُسجّل في الجدول بعد. عادةً ما يكون وقت انتهاء صلاحية معلومة تحويل العناوين حوالي 20 دقيقة من وقت إيداع المُدخل في جدول بروتوكول ARP.

عنوان IP	عنوان الماك	فترة العمر (TTL)
222.222.222.221	88-B2-2F-54-1A-0F	13:45:00
222.222.222.223	5C-66-AB-90-75-B1	13:52:00

الشكل[1] 5-18 جدول محتمل لبروتوكول ARP على العقدة 222.222.222.220.

افترض الآن أن العقدة 222.222.222.220 تريد إرسال وحدة بيانات معنونة بعنوان IP إلى عقدة أخرى على نفس الشبكة الفرعية. تحتاج عقدة الإرسال للحصول على عنوان الماك لعقدة الوجهة بمعلومية عنوان IP لتلك العقدة. تلك مهمة سهلة إذا كان جدول ARP لتحويل العناوين يتضمن المُدخل المطلوب لعنوان عقدة الوجهة. ولكن ماذا يحدث لو أن ذلك التحويل ليس مدرجاً حالياً في جدول ARP؟ بتحديد أكثر، افترض أن العقدة 222.222.222.220 بحاجة لإرسال وحدة بيانات إلى العقدة 222.222.222.222. في هذه الحالة تستخدم عقدة الإرسال بروتوكول تحويل العناوين لتحديد عنوان الماك. أولاً تُنشئ عقدة الإرسال رزمة خاصّة تسمى رزمة بروتوكول ARP. تتضمن تلك الرزمة عدّة حقول من بينها حقول لعناوين IP وعناوين الماك لكلٍّ من المُرسِل والمُستقبِل. تَستخدم رزم بروتوكول ARP نفس الصيغة للاستفسار والإجابة. الغرض من رزمة بروتوكول ARP للاستفسار هو سؤال كل العقد الأخرى الموصّلة على الشبكة الفرعية عن عنوان الماك المناظر لعنوان IP المراد تحديده.

عودةً إلى مثالنا الذي نحن بصدده، ترسل العقدة 222.222.222.220 رزمة استفسار ARP إلى الموائم مع إشارة تبين أن على الموائم إرسال الرزمة على عنوان الماك المخصص للإذاعة (أي FF-FF-FF-FF-FF-FF) . يغلّف الموائم رزمة استفسار ARP في إطار طبقة ربط البيانات، ويضع عنوان الإذاعة في حقل عنوان وجهة الإطار، ثم يرسل الإطار على الشبكة الفرعية. تذكّر التناظر الذي ذكرناه آنفاً

[1] يشار إلى بعض "الجداول" في الكتاب الأصلي بالأشكال، لذا تركنا الإشارة إليها "بالأشكال" من أجل عدم إحداث تغيير بتسلسل ترقيم الأشكال والجداول مما يسهل الرجوع للكتاب الأصلي (لمن أراد ذلك).

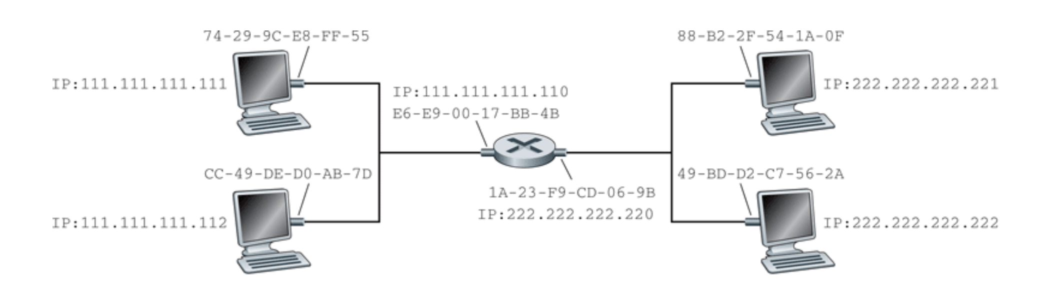
فيما يتعلق برقم الضمان الاجتماعي والعنوان البريدي حيث تمثل رزمة استفسار ARP شخصاً يصيح في غرفة مزدحمة بمقصورات المكاتب الصغيرة في شركةٍ ما (مثلاً شركة AnyCorp) قائلاً: "ما هو رقم الضمان الاجتماعي للشخص الذي عنوانه البريدي: مقصورة 13، غرفة 112، شركة AnyCorp، بالو ألتو، كاليفورنيا؟" يصل الإطار الذي يتضمن استفسار ARP إلى كل الموائمات الأخرى على الشبكة الفرعية، ونظراً لأن ذلك الإطار يحمل عنوان إذاعة، يقوم موائم كل عقدة بتمرير رزمة استفسار ARP التي استخلصها من الإطار إلى وحدة بروتوكول ARP الخاصة بتحويل العناوين على تلك العقدة. تفحص كل عقدة عنوان IP في رزمة استفسار ARP، وتقارنه بعنوان IP الخاص بها. تقوم العقدة التي تجد ذلك العنوان مطابقاً لعنوان IP الخاص بها بالرد على العقدة المستفسرة برزمة إجابة ARP تتضمن المطابقة المطلوبة بين عنوان IP المعروف وعنوان الماك المناظر. عندئذٍ يمكن للعقدة المستفسرة 222.222.222.220 تحديث جدول ARP لتحويل العناوين لديها ثم ترسل وحدة بيانات IP التي تود إرسالها بعد تغليفها في إطار طبقة ربط البيانات فيه عنوان الماك للوجهة هو نفسه عنوان الماك الخاص بالعقدة التي ردت على رزمة استفسار ARP السابقة.

هناك شيئان جديران بالملاحظة فيما يتعلق ببروتوكول ARP لتحويل العناوين. أولاً: في حين تُرسل رسالة استفسار ARP ضمن إطار إذاعة، تُرسل رسالة إجابة ARP ضمن إطار عادي (موجّه إلى عقدة واحدة). قبل مواصلة القراءة عليك أن تفكر في السبب وراء ذلك. ثانياً: يُعتبر بروتوكول ARP لتحويل العناوين من نوع "وصّل وشغّل" (plug-and-play)، بمعنى أن جدول ARP على العقدة يتم انشاؤه وتحديثه تلقائياً - أي لا يحتاج الأمر إلى تهيئته يدوياً بواسطة مدير النظام. وإذا حدث وفُصلت عقدة من الشبكة الفرعية، فإن المُدخل الخاص بها في تلك الجداول يتم حذفه في النهاية من جداول ARP على العقد المتبقية على الشبكة الفرعية.

إرسال وحدة بيانات إلى عقدة خارج نطاق الشبكة الفرعية

لعله يكون قد اتضح الآن كيف يعمل بروتوكول ARP لتحويل العناوين عندما تريد عقدة إرسال وحدة بيانات إلى عقدة أخرى تقع على نفس الشبكة الفرعية (تم تعريف الشبكة الفرعية بدقة في الجزء 4-4-2). دعنا الآن نناقش الحالة الأكثر تعقيداً عندما تريد عقدة على شبكة فرعية إرسال وحدة بيانات طبقة الشبكة إلى عقدة خارج نطاق الشبكة الفرعية (أي عبر موجّه إلى شبكة فرعية أخرى). سنناقش هذا الوضع في سياق الشكل 5-19، والذي يبين شبكة بسيطة تتكون من شبكتين فرعيتين موصلتين ببعضهما عن طريق موجّه.

هناك عدّة أشياء جديرة بالملاحظة فيما يتعلّق بالشكل 5-19. أولاً: هناك نوعان من العقد (مضيفات وموجّهات). لكل مضيف عنوان IP واحد وموائم واحد فقط. أما الموجّه - فكما لاحظنا في الفصل الرابع - فله عنوان IP لكل واجهة (interface) من واجهاته، ولكلٍّ منها هناك أيضاً موائم ووحدة بروتوكول ARP لتحويل العناوين. نظراً لأن الموجّه في الشكل 5-19 له واجهتان، فسيكون لديه عنوانان من عناوين IP، ووحدتا ARP، وموائمان. بالطبع يكون لكل موائم على الشبكة عنوان ماك خاص به.



الشكل 5-19 شبكتان فرعيتان موصّلتان عبر موجّه.

لاحظ أيضاً أن الشبكة الفرعية 1 لها العنوان 111.111.111/24، بينما الشبكة الفرعية 2 لها العنوان 222.222.222/24. وبالتالي تأخذ عناوين IP لكل الواجهات الموصلة بالشبكة الفرعية 1 الشكل 111.111.111.xxx، في حين تأخذ عناوين IP لكل الواجهات الموصلة بالشبكة الفرعية 2 الشكل 222.222.222.xxx.

لنناقش الآن كيف يقوم مضيف على الشبكة الفرعية 1 بإرسال وحدة بيانات إلى مضيف على الشبكة الفرعية 2. بالتحديد افترض أن المضيف 111.111.111.111 يريد إرسال وحدة بيانات IP إلى المضيف 222.222.222.222. يمرر المضيف المُرسِل وحدة البيانات إلى الموائم لديه كالعادة، غير أنه يتعين على المضيف المُرسِل أيضاً أن يبين للموائم عنوان ماك مناسب لوجهة تلك الوحدة. ما عنوان الماك الذي يمكن أن يستخدمه موائم المُرسِل؟ قد تتسرع بالتخمين بأن ذلك العنوان هو عنوان الماك لموائم مضيف الوجهة 222.222.222.222 أي 49-BD-D2-C7-56-2A، غير أن هذا التخمين خطأ للأسف! إذا استخدم موائم المُرسِل عنوان الماك ذلك، فلن يكترث أيٌّ من الموائمات على الشبكة الفرعية 1 برفع وحدة بيانات IP التي تصله إلى طبقة الشبكة الموجودة أعلاه لأن عنوان الماك لوجهة الإطار لن يطابق عنوان الماك لأيٍ منها. عندئذٍ ستموت وحدة البيانات تلك ويلفها النسيان.

أما إذا أمعنّا النظر في الشكل 5-19 فسنرى أنه لكي تتمكّن وحدة بيانات من الانتقال من العقدة 111.111.111.111 إلى عقدة على الشبكة الفرعية 2، ينبغي أن ترسل وحدة البيانات أولاً إلى واجهة الموجّه بعنوان IP 111.111.111.110. وهكذا يكون العنوان المناسب لوجهة الإطار هو عنوان الماك لواجهة الموجّه 111.111.111.110، أي E6-E9-00-17-BB-4B. ولكن كيف يحصل المضيف المُرسِل على عنوان الماك لـ 111.111.111.110؟ باستعمال بروتوكول ARP طبعاً! بمجرد حصول موائم المُرسِل على عنوان الماك هذا، يقوم بإنشاء إطار (يضم وحدة البيانات المعنونة إلى 222.222.222.222) ويرسل الإطار إلى الشبكة الفرعية 1. تكتشف واجهة الموجّه على الشبكة الفرعية 1 أن إطار طبقة ربط البيانات هذا موجّه إليها، فترفع الإطار إلى طبقة الشبكة على الموجّه. أخيراً انتقلت وحدة بيانات IP بنجاح من مضيف المصدر إلى الموجّه! لكنّ مهمتنا لم تنته بعد! لا يزال علينا نقل وحدة

البيانات من الموجّه إلى وجهتها النهائية. على الموجّه الآن تحديد الواجهة الصحيحة عليه والتي ينبغي إرسال وحدة البيانات إليها. كما بينّا في الفصل الرابع، يتم ذلك باستشارة جدول التوجيه الموجود على الموجّه. يُخبر جدول التوجيه الموجّه أن وحدة البيانات سترسل عن طريق واجهة الموجّه التي لها عنوان IP 222.222.222.220. تدفع تلك الواجهة بعد ذلك بوحدة البيانات إلى موائمها، والذي يقوم بدوره بتغليف وحدة البيانات في إطار جديد ويرسل الإطار إلى الشبكة الفرعية 2. في هذه المرّة يكون عنوان الماك لوجهة الإطار هو في الحقيقة عنوان الماك للوجهة النهائية للإطار. وكيف يحصل الموجّه على عنوان الماك لهذه الوجهة؟ من بروتوكول ARP طبعاً!

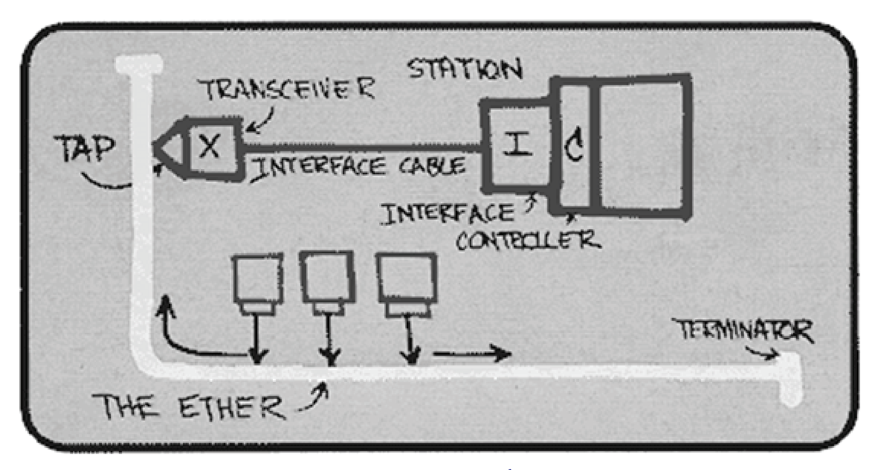
تم تعريف بروتوكول ARP للإيثرنت في طلب التعليقات RFC 826، كما توجد مقدمة لطيفة عن ARP في المقال التدريبي RFC 1180 عن بروتوكول TCP/IP. سوف نستكشف المزيد من تفاصيل بروتوكول ARP من خلال التمارين في نهاية الفصل.

5-5 شبكة الإيثرنت

لقد اكتسحت الإيثرنت تقريباً سوق شبكات البيانات المحلية السلكية. في الثمانينيات وأوائل التسعينيات واجهت الإيثرنت العديد من التحديات من التقنيات الأخرى لشبكات البيانات المحلية، بما في ذلك شبكات حلقة العلامة (token ring)، وشبكات واجهة البيانات الموزعة عبر الألياف الضوئية (FDDI)، وشبكات نمط النقل غير المتزامن (ATM). نجح البعض من تلك التقنيات في الاستحواذ على جزء من سوق الشبكات المحلية لبضع سنوات. غير أن الإيثرنت ومنذ اختراعها في أواسط السبعينيات واصلت نموها وتطورها وتمسّكت بمركزها المهيمن. واليوم تعتبر الإيثرنت إلى حد كبير أكثر تقنيات الشبكات المحلية انتشاراً، وهي مرشحة لتبقى كذلك في المُستقبل المنظور. قد يمكننا القول أن الإيثرنت كانت للشبكات المحلية بمثابة الإنترنت للشبكات العالمية.

هناك العديد من الأسباب التي ساهمت في نجاح الإيثرنت. أولاً: كانت الإيثرنت أول شبكة محلية سريعة قُدِّر لها الانتشار على نطاق واسع. ونظراً

لانتشارها المبكر، ألِفَ مشرفو الشبكات الإيثرنت عن كَثَب – بعجائبها والتواءاتها – ومن ثم كانوا يعارضون التحوّل إلى تقنيات الشبكات المحلية الأخرى عند ظهورها على الساحة. ثانياً: كانت التقنيات الأخرى – مثل: token ring، FDDI، ATM – أكثر تعقيداً وأغلى كلفةً من الإيثرنت، الأمر الذي ثبّط عزيمة مشرفي الشبكات أكثر عن ترك الإيثرنت والتحوّل إلى تلك التقنيات الجديدة. ثالثاً: كان السبب الأكثر إقناعاً للتحوُّل إلى تقنية شبكة محلية أخرى (مثل: FDDI أو ATM) عادةً هو المعدّلات الأعلى لإرسال البيانات التي توفرها تلك التقنيات الجديدة، ولكن الإيثرنت كانت دائماً تستبسل في المقاومة منتجةً نسخاً جديدة تعمل بمعدلات إرسال تساوي أو تتجاوز تلك المعدّلات. وفي بداية التسعينيات ظهرت الإيثرنت المحوّلة (switched Ethernet)، مما أدى إلى زيادةٍ أكبر في معدلات الإرسال الفعلية. وأخيراً: نظراً لزيادة شعبية الإيثرنت وانتشارها، أصبحت أجهزة الإيثرنت (وبخاصة الموائمات والمحوّلات) سلعاً رائجة ورخيصة جداً.

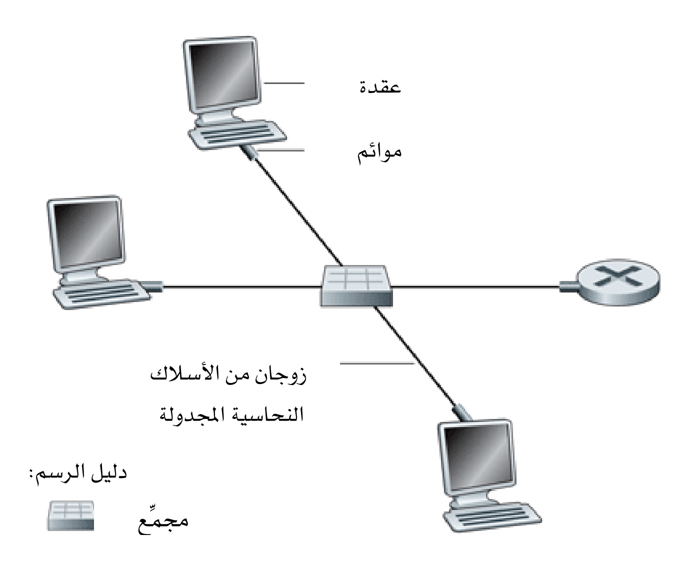


الشكل 5-20 تصميم ميتكالف الأصلي لمعيار BASE510 لشبكة الإيثرنت، والذي تضمّن كبل واجهة يصل موائم الإيثرنت بجهاز إرسال واستقبال خارجي.

اختُرعت شبكة الإيثرنت المحلية الأصلية في منتصف السبعينيات من قِبَل بوب ميتكالف وديفيد بوجز. يبين الشكل 5-20 رسماً تخطيطياً لميتكالف لهذا الاختراع. ستلاحظ في الشكل أن شبكة الإيثرنت المحلية الأصلية كانت تستخدم ناقلاً محورياً (coaxial bus) لربط العقد الموصّلة بالشبكة. في الواقع استمرت تقنية الناقل المحوري لطبوغرافية شبكة الإيثرنت على مدار الثمانينيات وحتى أواسط التسعينيات. جديرٌ بالذكر أن الإيثرنت بهيئة الناقل المحوري تمثل شبكة محلية بقناة إذاعة مشتركة، حيث تنتقل كل الإطارات المُرسَلة إلى كل الموائمات الموصّلة بالناقل وتتم معالجتها من قِبَلها.

بنهاية التسعينيات كانت معظم الشركات والجامعات قد استبدلت شبكاتها المحلية بتجهيزات إيثرنت تستخدم طبوغرافية النجمة (star topology) التي أساسها مجمِّع (hub). كما هو مبين في الشكل 5-21، في مثل هذه الترتيبة توصّل المضيفات (والموجّه) مباشرة إلى مجمِّع بزوج من الأسلاك النحاسية المجدولة. المجمِّع هو أداة تابعة للطبقة المادية تتعامل مع البتات المفردة وليس الإطارات. عندما يتلقى المجمِّع بتاً (يمثّل 0 أو 1) من إحدى واجهاته فإنه يقوم ببساطة بتكوين البت من جديد برفع طاقة إشارته الكهربية، ثم يرسله إلى كل الواجهات الأخرى. وهكذا فإن الإيثرنت بترتيبة نجمية ومجمِّع في المركز لاتزال شبكة إذاعة محلية. بتحديد أكثر إذا استلم المجمِّع إطارات من واجهتين مختلفتين في نفس الوقت سيحدث اصطدام، وسيتعين على العُقد التي أنشأت تلك الإطارات إعادة إرسالها.

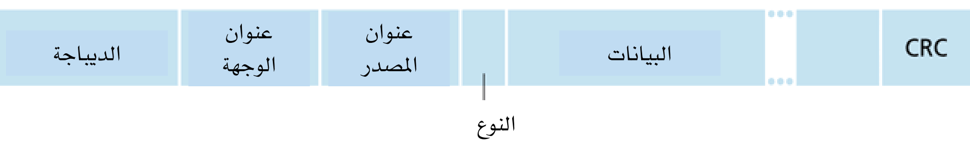
في بداية القرن الجديد طرأ على الإيثرنت تطوير رئيس آخر. واصلت تجهيزات الإيثرنت استخدام طبوغرافية النجمة، ولكن مع استبدال المجمِّع الموجود في المركز بمحوّل (switch). سنفحص الإيثرنت المحوّلة بتفصيل أكثر لاحقاً في هذا الفصل. نكتفي الآن بالقول بأن المحوّل لا يمنع الاصطدام فقط، بل ويعتبر كذلك مثالاً أصيلاً لمحوّل الرزم بأسلوب "خزّن ومرّر" (store-and-forward). ولكن على خلاف الموجّه الذي يعمل حتى طبقة 3 في رصة البروتوكولات، فإن المحوّل يعمل حتى طبقة 2 فقط.



الشكل 5-21 طبوغرافية النجمة للإيثرنت. يتم توصيل العقد بعضها ببعض عن طريق مجمِّع.

5-5-1 صيغة إطار الإيثرنت

يمكننا تعلّم الكثير عن الإيثرنت بفحص إطار الإيثرنت والمبيَّن في الشكل 5-22. لإضفاء طابع واقعي على هذه المناقشة حول إطارات الإيثرنت، دعنا نأخذ في الاعتبار إرسال وحدة بيانات IP من مضيف إلى مضيف آخر يقع على نفس شبكة الإيثرنت المحلية (على سبيل المثال شبكة الإيثرنت المبينة في الشكل 5-21). رغم أن حمولة إطار الإيثرنت في حالتنا هذه هي وحدة بيانات IP، إلا أننا نذكر هنا بشكلٍ عابر أن إطار الإيثرنت يمكن أيضاً أن يحمل رزماً لأنواع أخرى من طبقة الشبكة. افترض أن موائم الإرسال A له عنوان الماك AA-AA-AA-AA-AA-AA وموائم الاستلام B له عنوان الماك BB-BB-BB-BB-BB-BB. يقوم موائم الإرسال بتغليف وحدة بيانات IP ضمن إطار إيثرنت ويدفع به لأسفل إلى الطبقة المادية. يستلم موائم الاستلام الإطار من الطبقة المادية أسفله، ويستخلص وحدة بيانات IP منه، ثم يمررها إلى طبقة الشبكة أعلاه. في هذا السياق دعنا الآن نفحص الحقول الستّة التي يتألف منها إطار الإيثرنت كما هو موضح في الشكل 5-22:



الشكل 5-22 صيغة إطار الإيثرنت.

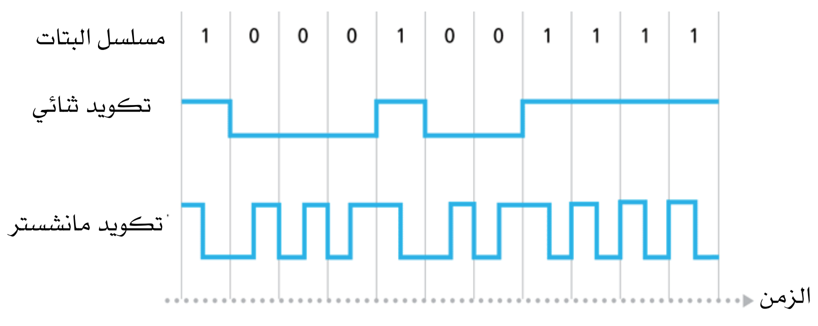
- حقل البيانات (يتكوّن من 46 بايتاً إلى 1500 بايت): يحمل هذا الحقل وحدة بيانات IP. يبلغ حجم وحدة الإرسال القصوى (Maximum Transmission Unit (MTU)) للإيثرنت 1500 بايت، وهذا يعني أنه إذا تجاوزت وحدة بيانات IP 1500 بايت فإن المضيف يضطر لتجزيء حزمة البيانات كما هو موضحٌ في الجزء 4-4-1. أما الحد الأدنى لحقل البيانات فهو 46 بايتاً، وهذا يعني أنه إذا كانت وحدة بيانات IP أقل من 46 بايتاً فإنه ينبغي "حشو" حقل البيانات لملئه حتى 46 بايتاً. في حالة استخدام الحشو (stuffing)، تتضمن البيانات التي تُرفع إلى طبقة الشبكة الحشو بالإضافة إلى وحدة بيانات IP الأصلية. في هذه الحالة تستخدم طبقة الشبكة حقل الطول في ترويسة وحدة بيانات IP لإزالة الحشو.
- عنوان الوجهة (6 بايتات): يحتوي هذا الحقل على عنوان الماك لموائم الوجهة، أي BB-BB-BB-BB-BB-BB في المثال الذي نحن بصدده. عندما يتلقى موائم B إطار إيثرنت يحمل في حقل عنوان الوجهة BB-BB-BB-BB-BB-BB أو عنوان الماك المخصص لعملية الإذاعة، فإنه يمرر محتويات حقل البيانات الموجود في الإطار إلى طبقة الشبكة أعلاه. أما إذا تلقى إطاراً بأي عنوان ماك آخر فإنه يستبعد ذلك الإطار ولا يعيره أي اهتمام.
- عنوان المصدر (6 بايتات): يحتوي هذا الحقل على عنوان الماك للموائم الذي يرسل الإطار على شبكة البيانات المحلية. في مثالنا الحالي يكون هذا العنوان AA-AA-AA-AA-AA-AA.
- النوع (بايتان): يسمح حقل النوع للإيثرنت بالقيام بعملية التجميع (multiplexing) لبروتوكولات مختلفة لطبقة الشبكة. لفهم هذه الحقيقة

تذكّر أن المضيفات يمكن أن تستخدم بروتوكولات أخرى لطبقة الشبكة بالإضافة إلى بروتوكول IP. في الواقع قد يدعم مضيف بعينه عدة بروتوكولات لطبقة الشبكة، حيث يستخدم المضيف بروتوكولات مختلفة مع التطبيقات المختلفة. لهذا السبب عندما يصل إطار إيثرنت إلى الموائم B يحتاج هذا الموائم لمعرفة بروتوكول طبقة الشبكة الذي يجب أن يمرر (أي يوزع demultiplex) محتويات حقل البيانات ضمن ذلك الإطار إليه. لكلٍّ من بروتوكول IP وغيره من بروتوكولات طبقة الشبكة الأخرى (على سبيل المثال: Novell ، IPX، وAppleTalk) رقم النوع المعياري الذي يميِّزه. علاوةً على ذلك فإن بروتوكول تحويل العناوين ARP (والذي تناولناه في الجزء السابق) له أيضاً رقم النوع الخاص به. لاحظ أن حقل النوع يشبه حقل البروتوكول في وحدة بيانات طبقة الشبكة وحقل رقم المنفذ في قطعة طبقة النقل؛ والغرض من كل تلك الحقول وصل بروتوكول في طبقة ما ببروتوكول في طبقة تعلوها.

- شفرة فحص الفائض الدوري CRC (4 بايتات): كما تقدّم في الجزء 5-2-3، الغرض من حقل شفرة CRC هو تمكين موائم الاستقبال ـ الموائم B ـ من اكتشاف ما إذا كانت هناك أي أخطاء قد طرأت على الإطار أثناء انتقاله من موائم الإرسال، أي ما إذا كانت أيٌّ من بتات الإطار قد تغيرت (1 تحوّل إلى 0 أو 0 تحوّل إلى 1). تتضمّن أسباب وقوع أخطاء في البتات: الاضمحلال في قوة الإشارة، ووجود طاقة كهرومغناطيسية محيطة تتسرب إلى كبلات الإيثرنت وبطاقات المواءمة. يتم اكتشاف الأخطاء كالتّالي: عندما ينشئ مضيف A إطار الإيثرنت لإرساله، يقوم بتعيين قيمة حقل CRC بالإطار، والتي تحسب كدالةٍ في كل بتات الإطار الأخرى ما عدا بتات الديباجة (الاستهلال) (preamble). وعندما يستلم مضيف B الإطار يطبّق نفس الدالة على نفس الجزء من الإطار الذي وصله ليرى ما إذا كانت النتيجة مساويةً للقيمة الموجودة في حقل CRC بالإطار. يطلق على هذه العملية في مضيف الاستقبال تدقيق CRC. إذا كانت نتيجة ذلك الفحص سلبية (أي كانت

نتيجة تطبيق الدالة على بقية الإطار لا تساوي محتويات حقل CRC) فإن المضيف B يدرك أن خطأً قد طرأ على الإطار.

- الديباجة (8 بايتات): يبدأ إطار الإيثرنت بحقل ديباجة طوله 8 بايتات. البايتات السبع الأولى في الديباجة لها نفس القيمة وهي 10101010، في حين يكون البايت الأخير 10101011. تستخدم البايتات السبع الأولى لـ"إيقاظ" موائمات الاستقبال ولتحقيق التزامن بين ساعات التوقيت لديها وساعة التوقيت لدى المُرسِل. لماذا يمكن أن تكون الساعات غير متزامنة؟ تذكّر أن الموائم A يهدف لإرسال الإطار بمعدّل 10 ميجابت/ثانية، أو 100 ميجابت/ثانية، أو 1 جيجابت/ثانية حسب نوع شبكة الإيثرنت المحلية. ومع ذلك فنظراً لأنه لا يوجد شيءٌ مثاليٌ في هذا العالم، فلن يرسل الموائم A الإطار بنفس معدل الإرسال المستهدف بالضبط، بل سيكون هناك دائماً بعض الانحراف عن هذا المعدّل – انحراف لا يُعرف مقداره مسبقاً لدى الموائمات الأخرى على الشبكة المحلية. بوسع موائم الاستقبال أن يحقق المطابقة المطلوبة مع ساعة موائم الإرسال A ببساطة بالمطابقة على بتات البايتات السبع الأولى من الديباجة. أما البتان الأخيران من بتات البايت الثامن في الديباجة (بقيمة 1 لكلٍّ منهما) فتنبهان الموائم B إلى أن "الأشياء المهمة على وشك الوصول". عندما يرى المضيف B البتين المتتاليين بقيمة 1، فإنه يدرك أن البايتات الست القادمة هي عنوان الوجهة. يمكن لموائمٍ ما أن يعرف أن إطاراً قد انتهى ببساطة بملاحظة غياب التيار على الوصلة المادية.



الشكل 5-23 تكويد مانشستر.

تستخدم الإيثرنت إرسالاً في حيز التردد الأصلي (baseband transmission)، بمعنى أن الموائم يرسل الإشارة الرقمية مباشرة إلى قناة الإذاعة (أي لا تنقل بطاقة الواجهة الإشارة إلى نطاقٍ ترددي آخر كما يحدث في أنظمة خط المشترِك الرقمي غير المتماثل (ADSL) ونظام مودم الكبل (cable modem). تستخدم العديد من تقنيات الإيثرنت (مثلاً 10BASE-T) تكويد مانشستر (Manchester coding)، كما هو مبين في الشكل 5-23. في هذا الأسلوب تتضمن إشارة كل بت انتقالاً في مستوى الإشارة: يُمثّل البت 1 بانتقال من أعلى إلى أسفل بينما يُمثّل البت 0 بانتقال من أسفل إلى أعلى. يرجع السبب في استخدام كود مانشستر إلى أن ساعات التوقيت لدى موائمات الإرسال والاستقبال تكون غير متزامنة تماماً في واقع الأمر. يساعد وجود انتقال في الإشارة دائماً في منتصف كل بت مضيفَ الاستقبال في أن يزامن ساعته مع ساعة مضيف الإرسال. بمجرد تحقيق ذلك التزامن لساعة موائم الاستقبال سيكون بوسع المُستقبِل تحديد موقع كل بت يتم استقباله وتعيين ما إذا كانت قيمته 1 أو 0. يُلاحَظ أن عملية تكويد البيانات بكود مانشستر تتم في الطبقة المادية وليس في طبقة ربط البيانات، ولكننا آثرنا الإلماح إليها سريعاً هنا لأنها تُستخدم على نطاق واسع في الإيثرنت.

الخدمة اللاتوصيلية غير الموثوقة

توفر كل تقنيات الإيثرنت خدمة لاتوصيلية (connectionless) لنقل البيانات لطبقة الشبكة. أي أنه عندما يريد الموائم A إرسال وحدة بيانات إلى الموائم B فإنه يغلّف وحدة البيانات في إطار إيثرنت، ويرسل الإطار على الشبكة المحلية، دون أن يسبق ذلك أي إجراءات مصافحة (handshaking) لإنشاء توصيلة مع الموائم B. إن هذه الخدمة اللاتوصيلية في الطبقة 2 تشبه خدمة IP لنقل وحدات البيانات في الطبقة 3 وخدمة UDP اللاتوصيلية في الطبقة 4.

دراسة حالة (Case Study)

بوب ميتكالف والإيثرنت

كطالب دكتوراه في جامعة هارفارد في أوائل السبعينيات، عمل بوب ميتكالف على شبكة أربانت (ARPAnet) في معهد ماسوشيستس للتكنولوجيا (MIT). ومن خلال دراساته اطّلع ميتكالف أيضاً على جهود أبرامسون في مجال تطوير بروتوكول ألوها وبروتوكولات الوصول العشوائي. وبعد إكماله دراسة الدكتوراه وقبل التحاقه مباشرةً بوظيفته الجديدة بمركز أبحاث زيروكس (Xerox) في بالو ألتو (Xerox PARC)، قام ميتكالف بزيارة أبرامسون وزملائه بجامعة هاواي لمدة ثلاثة أشهر، حيث اطلع عن كثب على شبكة ألوهانت. في مركز أبحاث Xerox PARC، تعامل ميتكالف مع حاسبات الألتو، والتي كانت تعتبر لأكثر من سبب الجيل المتقدم الذي سبق ظهور الحاسبات الشخصية في الثمانينيات. أيقن ميتكالف بالحاجة لتشبيك تلك الحاسبات بطريقة قليلة الكلفة. وهكذا بنى ميتكالف على معرفته بشبكات الأربانت والألوهانت وبروتوكولات الوصول العشوائي، ليتمكن مع زميله ديفيد بوجز من اختراع الإيثرنت.

عملت شبكة الإيثرنت الأصلية بمعدّل إرسال قدره 2.94 ميجابت/ثانية وربطت مضيفات وصل عددها إلى 256 مضيف فصلت بينها مسافات وصلت إلى ميل واحد. نجح ميتكالف وبوجز في تمكين أغلب الباحثين في معهد أبحاث Xerox PARC من الاتصال ببعضهم البعض من خلال حاسبات ألتو لديهم. بعد ذلك صاغ ميتكالف تحالفاً بين شركات Xerox و Digital و Intel لتأسيس الإيثرنت كشبكة معيارية بمعدّل إرسال 10 ميجابت/ثانية، والتي صدّقت عليها منظمة IEEE. لم تهتم Xerox كثيراً بالتطبيقات التجارية للإيثرنت. وفي عام 1979 أسس ميتكالف شركته الخاصة Com3، والتي طوّرت تقنيات الربط بالشبكات بما في ذلك تقنية الإيثرنت واهتمت بتطبيقاتها التجارية. في أوائل الثمانينيات طوّرت Com3 وسوّقت بطاقات الإيثرنت لحاسب IBM الشخصي ذائع الصيت آنذاك. وفي عام 1990 ترك ميتكالف Com3 عندما كان لديها 2,000 موظف وعائداتها 400 مليون دولار.

أيضاً توفر كل تقنيات الإيثرنت خدمة غير موثوقة (unreliable) لطبقة الشبكة. وبتحديدٍ أكثر عندما يستلم الموائم B إطاراً من الموائم A فإنه يُخضع الإطار لتدقيق CRC لاكتشاف الخطأ، ولكنه لا يرسل إشعار استلام عندما يجتاز الإطار عملية الفحص تلك، ولا إشعاراً سلبياً عندما لا يجتاز الإطار هذا الفحص. وعليه فعندما يفشل إطار في اجتياز تدقيق CRC لاكتشاف الأخطاء فإن الموائم B يكتفي فقط بإهمال ذلك الإطار. وهكذا لن تكون لدى الموائم A أي فكرة عما إذا كان إطاره الذي أرسله قد وصل إلى الموائم B واجتاز فحص اكتشاف الأخطاء أم لا. إن غياب إمكانيات النقل الموثوق (في طبقة ربط البيانات) تساعد في جعل الإيثرنت بسيطة ورخيصة الكلفة، غير أن ذلك يعني في المقابل أن سلسلة وحدات البيانات التي تمرّر إلى طبقة الشبكة يمكن أن تحدث بها فجوات.

إذا كانت هناك فجوات بسبب إهمال بعض إطارات الإيثرنت المعطوبة فهل يرى التطبيق على المضيف B تلك الفجوات هو الآخر؟ كما رأينا في الفصل الثالث يعتمد هذا على ما إذا كان التطبيق يستخدم بروتوكول UDP أو بروتوكول TCP. في حالة استخدام بروتوكول UDP فإن التطبيق على المضيف B سيلحظ فعلاً وجود فجوات في البيانات. وفي المقابل إذا استخدم التطبيق بروتوكول TCP فإن المضيف B لن يُرسل إشعارات باستلام البيانات المُرسِلة في الإطارات التي تم إهمالها، مما يجعل بروتوكول TCP على المضيف A يعيد إرسال تلك البيانات من جديد. لاحظ أنه عندما يعيد TCP إرسال البيانات فستعود البيانات في النهاية إلى موائم الإيثرنت الذي أهملها في السابق. وهكذا فإن الإيثرنت تعيد إرسال البيانات، ولكنها لا تدري ما إذا كانت تنقل وحدة بيانات جديدة محمّلة ببيانات جديدة أو وحدة بيانات تحتوي على بيانات سبق إرسالها مرة واحدة على الأقل.

5-5-2 بروتوكول الوصول المتعدد للإيثرنت: الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام (CSMA/CD)

عندما تُوصّل العقد فيما بينها عن طريق مجمِّع (hub) (في مقابل محوّل طبقة ربط البيانات (switch))، كما هو مبين في الشكل 5-21، تكون شبكة الإيثرنت

شبكة إذاعة محلية بحق - بمعنى أنه عندما يرسل موائم إطاراً، فإن كل الموائمات الموصلة على شبكة البيانات المحلية تتلقى ذلك الإطار. نظراً لأن الإيثرنت يمكن أن تستخدم أسلوب الإذاعة، فإنها تحتاج ابتداءً إلى نظام وصول متعدد. تستخدم الإيثرنت البروتوكول الشهير للوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام (CSMA/CD). تذكّر من استعراضنا لذلك البروتوكول في الجزء 5-3 أن بروتوكول CSMA/CD يعمل كالتالي:

1. يمكن لموائم البدء في الإرسال في أي وقت يشاء، بمعنى أن البروتوكول لا يستخدم مفهوم الشرائح الزمنية.
2. لن يرسل الموائم إطاراً أبداً بمجرد إحساسه بأن موائماً آخر يرسل حالياً، بمعنى أن الموائم يستخدم أسلوب الإنصات للناقل.
3. يقوم الموائم بقطع إرساله بمجرد اكتشافه أن موائماً آخر يرسل أيضاً، بمعنى أن الموائم يستخدم أسلوب اكتشاف الاصطدام.
4. قبل محاولة إعادة الإرسال يقوم الموائم بالانتظار لوقت عشوائي عادةً ما يكون صغيراً مقارنةً بالوقت اللازم لإرسال إطار.

توفر تلك الآليات لبروتوكول CSMA/CD أداءً أفضل بكثير من أداء بروتوكول ألوها الشرائحي في بيئة الشبكة المحلية. في الحقيقة عندما يكون تأخير الانتقال الأقصى بين العقد صغيراً جداً، فإن كفاءة بروتوكول CSMA/CD يمكن أن تقترب من 100٪. لكن ينبغي ملاحظة أن الآليات رقم 2 و3 المذكورة أعلاه تتطلب من كل موائم إيثرنت أن يكون قادراً على: (1) الإحساس بما إذا كان هناك موائم آخر يرسل، و(2) اكتشاف وقوع اصطدام أثناء عملية الإرسال. تؤدّي موائمات الإيثرنت هاتين المهمّتين بقياس مستويات الجهد الكهربائى (الفولطية) قبل وأثناء الإرسال.

ينفّذ كل موائم بروتوكول CSMA/CD بدون تنسيق محدد مع الموائمات الأخرى على الإيثرنت. ضمن موائم بعينه يعمل بروتوكول CSMA/CD كالتّالي:

1. يحصل الموائم على وحدة بيانات من طبقة الشبكة، فينشئ إطار إيثرنت، ويضع الإطار في المخزن المؤقت على الموائم.

2. إذا أحس الموائم أن القناة شاغرة (أي أنه لا تدخل طاقة إشارة الموائم من القناة لفترة تبلغ مدة إرسال 96 بتاً)، فإنه يبدأ في إرسال الإطار. إذا أحس الموائم أن القناة مشغولة، فإنه ينتظر إلى أن تختفي أي طاقة إشارة (زائد مدة إرسال 96 بتاً) وبعد ذلك يبدأ في إرسال الإطار.
3. أثناء إرسال الإطار، يراقب الموائم القناة لاكتشاف وجود طاقة إشارة قادمة من الموائمات الأخرى. إذا تمكّن الموائم من إرسال الإطار كاملاً بدون اكتشاف طاقة إشارة من الموائمات الأخرى فإنه يكون قد نجح في إرسال ذلك الإطار.
4. إذا اكتشف الموائم طاقة إشارة من الموائمات الأخرى أثناء قيامه بالإرسال، فإنه يتوقف عن إرسال إطاره ويرسل بدلاً من ذلك إشارة تشويش طولها 48 بتاً.
5. بعد قطع الإرسال (وإرسال إشارة التشويش) يدخل الموائم مرحلة تراجع أسّي (exponential backoff). بالتحديد عندما تواجه عملية إرسال إطارٍ بعينه الاصطدام رقم n على التوالي، فإن الموائم يختار قيمة عشوائية للمتغير K من بين القيم $(0, 1, 2, \ldots, 2^m -1)$، حيث $m = \min(n, 10)$. ينتظر الموائم مدة إرسال $512 \times K$ بتاً وبعدها يعود للخطوة 2.

من المفيد هنا ذكر بضعة تعليقات حول بروتوكول CSMA/CD. إن الغرض من بث إشارة التشويش التأكد من أن كل موائمات الإرسال الأخرى قد أدركت وجود الاصطدام. دعنا نأخذ هذا المثال. افترض أن الموائم A يبدأ في إرسال إطار، ولكن مباشرةً قبل وصول إشارة إطار A إلى B يبدأ الموائم B في الإرسال. لذا يكون B قد أرسل فقط بضعة بتات عندما يقطع إرساله. هذه البتات القليلة ستنتقل بالفعل إلى A، ولكنّها قد لا تشكّل طاقة إشارة كافية لتمكين A من اكتشاف وجود الاصطدام . للتأكد من أن A يكتشف الاصطدام (لكي يقوم هو الآخر بقطع إرساله)، يقوم B بإرسال إشارة تشويش طولها 48 بتاً.

لنأخذ في الاعتبار الآن خوارزمية التراجع الأسّي. أول ما نلاحظه هنا هو أن وقت البت (أي الوقت الذي يستغرقه إرسال بت واحد) قصيرٌ جداً، فعلى إيثرنت

سرعتها 10 ميجابت/ثانية يكون وقت البت 0.1 ميكروثانية. دعنا الآن نأخذ هذا المثال: افترض أن موائماً يحاول إرسال إطارٍ للمرة الأولى ولكنه يكتشف اصطداماً أثناء الإرسال. يختار الموائم $K = 0$ باحتمال 0.5 أو يختار $K = 1$ باحتمال 0.5. إذا اختار الموائم $K = 0$، فإنه يقفز فوراً لخطوة 2 بعد إرسال إشارة التشويش. إذا اختار الموائم $K = 1$ فإنه ينتظر 51.2 ميكروثانية قبل العودة لخطوة 2. بعد اصطدامٍ ثانٍ، يتم اختيار K باحتمالات متساوية من بين القيم (0، 1، 2، 3). بعد ثلاثة اصطدامات، يتم اختيار K باحتمالات متساوية من بين القيم (0، 1، 2، 3، 4، 5، 6، 7). بعد عشرة اصطدامات أو أكثر، يتم اختيار K باحتمالات متساوية من بين القيم (0، 1، 2، ...، 1023). وهكذا فإن حجم مجموعة الأعداد الذي تُختار منه قيمة K ينمو تصاعدياً مع عدد الاصطدامات (حتى $n = 10$)، ولهذا السبب تُدعى خوارزمية التراجع في الإيثرنت خوارزمية أُسّية.

يفرض معيار الإيثرنت حدوداً قصوى على المسافة بين أي عقدتين على الشبكة. تضمن تلك الحدود أنه إذا اختار الموائم A قيمة منخفضة للمتغير K عن كل الموائمات الأخرى التي اشتركت معه في الاصطدام، فإن الموائم A يكون بوسعه إرسال إطاره بدون مواجهة اصطدام جديد. سنستكشف تلك الخاصية بتفصيل أكثر في تمارين نهاية الفصل.

لماذا نستخدم تراجعاً أسّياً؟ لِمَ لا نختار K على سبيل المثال من بين 0، 1، 2، 3، 4، 5، 7، 6 بعد كل اصطدام؟ السبب أنه عندما يواجه موائم أول اصطدام له فإنه لا يدري كم عدد الموائمات المتورطة في ذلك الاصطدام. إذا كان هناك عدد صغير من تلك الموائمات، فإنه يكون من الحكمة اختيار K من مجموعة قليلة من القيم الصغيرة. وفي المقابل إذا كان هناك العديد من الموائمات المشتركة في الاصطدام، فمن الأفضل اختيار K من مجموعة أكبر من القيم الأكثر تفاوتاً (لماذا؟). لاحظ أنه بزيادة حجم المجموعة بعد كل اصطدام، يتكيّف الموائم بشكلٍ ملائم مع تلك السيناريوهات المختلفة.

نلاحظ هنا أيضاً أنه في كل مرة يقوم موائم بإنشاء إطار جديد للإرسال، فإنه يقوم بتنفيذ خوارزمية CSMA/CD المبينة أعلاه. وبشكلٍ خاص لا يأخذ الموائم في اعتباره أي اصطدامات ربما تكون قد وقعت في الماضي القريب. لذا فقد يتمكن موائم لديه إطار جديد من الانسلال بسرعة والنجاح في إرسال الإطار بينما تكون عدّة موائمات أخرى في حالة التراجع الأسّي.

كفاءة الإيثرنت

عندما يكون لدى عقدة واحدة فقط إطار للإرسال، يمكن لتلك العقدة أن ترسل بمعدل الإرسال الكامل لتقنية الإيثرنت المستخدمة (مثلاً 10 ميجابت/ثانية، أو 100 ميجابت/ثانية، أو 1 جيجابت/ثانية). ولكن إذا كان لدى العديد من العقد إطارات للإرسال، فإن معدل الإرسال الفعّال للقناة يمكن أن يكون أقل من ذلك بكثير. نُعرّف هنا كفاءة الإيثرنت (Ethernet efficiency) على أنها الكسر من الوقت على المدى البعيد الذي يتم فيه إرسال الإطارات على القناة بدون اصطدامات، وذلك في وجود عدد كبير من العقد النشطة لدى كلٍّ منها عدد كبير من الإطارات للإرسال. للحصول على معادلة تقريبية تمثل كفاءة الإيثرنت، افترض أن d_{prop} تمثل الوقت الأقصى الذي تستغرقه طاقة الإشارة للانتقال بين أي وصلتين، وd_{trans} الوقت اللازم لإرسال إطار إيثرنت له أقصى حجم ممكن (تقريباً 1.2 ميللي ثانية للإيثرنت بسرعة 10 ميجابت/ثانية). يقع اشتقاق كفاءة الإيثرنت خارج نطاق هذا الكتاب (انظر [Lam 1980] و[Bertsekas 1991])، ولكننا سنكتفي هنا ببساطة بذكر التقريب التالي للكفاءة:

$$Efficiency = \frac{1}{1 + 5d_{prop}/d_{tran}}$$

نرى من هذه المعادلة أنه عندما تقترب d_{prop} من 0، فإن الكفاءة تقترب من 1. إن هذا يتفق مع نظرتنا البدهية، حيث إنه إذا كان تأخير الانتقال صفراً، فإن العقد المتصادمة ستتوقف عن إرسالها فوراً بدون إهدار لوقت القناة. أيضاً كلما أصبحت d_{trans} كبيرة جداً، تقترب الكفاءة من 1. هذا بدهي أيضاً لأنه عندما يتمكن إطار

من الاستحواذ على القناة، فإنه سيتمسّك بها لوقت طويل جداً، أي أن القناة ستعمل عملاً منتجاً أغلب الوقت.

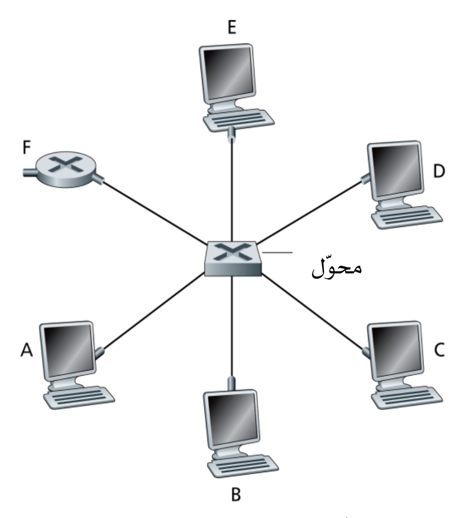
5-5-3 تقنيات الإيثرنت

في مناقشتنا أعلاه، كنا نشير إلى الإيثرنت كما لو كانت بروتوكولاً معيارياً واحداً. غير أنه في الواقع تأخذ الإيثرنت العديد من الأشكال المختلفة، وتستخدم بعض الاختصارات المحيّرة أحياناً مثل: 10BASE-T، 10BASE-2، 100BASE-T، 1000BASE-LX، 10GBASE-T. تم اعتماد هذه والعديد غيرها من تقنيات الإيثرنت الأخرى كمعايير قياسية على مرّ السنين من مجموعات العمل كمجموعة IEEE 802.3 CSMA/CD [IEEE 802.3 2007]. رغم أن هذه الاختصارات قد تبدو محيّرة بعض الشيء، إلا أنها تحمل في طياتها قدراً كبيراً من المنطق. فالجزء الأول من الاختصار يشير إلى سرعة الإرسال للمعيار: فالأرقام 10، و100، و1000، و10G تمثل 10 ميجابت/ثانية، و100 ميجابت/ثانية، و1 جيجابت/ثانية، و10 جيجابت/ثانية على الترتيب. تشير كلمة BASE إلى أن الإيثرنت ترسل في حيز التردد الأصلي (baseband)، بمعنى أن وسط الانتقال المادي يحمل فقط حركة بيانات الإيثرنت. كل معايير 802.3 تعرّف إيثرنت بحيز التردد الأصلي. يدل الجزء الأخير من الاختصار على وسط الانتقال المادي نفسه، فالإيثرنت تمثل مواصفات لكلٍّ من طبقة ربط البيانات والطبقة المادية، وهي تستخدم تشكيلة من الأوساط المادية لنقل الإشارات بما في ذلك الكبل المحوري، وأسلاك النحاس، والألياف الضوئية. عموماً ترمز T لزوج مجدول من الأسلاك النحاسية.

تاريخياً كانت الإيثرنت في البداية تقترن في المخيّلة على أنها قطعة (segment) من كبل محوري كما هو مبين في الشكل 5-20. تصف المعايير الأولى 10BASE-2 و10BASE-5 شبكة إيثرنت بمعدل إرسال قدره 10 ميجابت/ثانية على نوعين من أنواع الكبل المحوري، بطول يصل إلى 200 متر و500 متر على الترتيب. يمكن تغطية مسافات أطول من ذلك باستخدام مُكرِّر (repeater)، وهو جهاز يعمل في الطبقة المادية حيث يتلقى إشارة من ناحية المدخل ويعيد توليدها مجدداً على ناحية

المخرج. إن الكبل المحوري، كما يبدو في الشكل 5-20، يطابق بشكلٍ جيد مفهومنا عن الإيثرنت كوسط إذاعة- حيث يتم استقبال كل الإطارات التي ترسلها إحدى الواجهات بواسطة كل الواجهات الأخرى، ويحل بروتوكول الإيثرنت CSMA/CD مشكلة الوصول المتعددة بشكلٍ رائع. ما علينا إلا أن نربط العقد بالكبل ببساطة، فنحصل على شبكة بيانات محلية!

لقد مرّت الإيثرنت عبر سلسلة من التطورات على مرّ السنين، وإيثرنت اليوم تختلف كثيراً عن التصاميم الأصلية بترتيبة ناقل مشترك يأخذ شكل كبل محوري. في أكثر تجهيزات إيثرنت اليوم، توصّل العقد إلى محوّل (switch) عن طريق وصلات نقطة إلى نقطة مصنوعة من أسلاك النحاس المجدولة أو الألياف الضوئية كما هو مبيّن في الشكل 5-24.



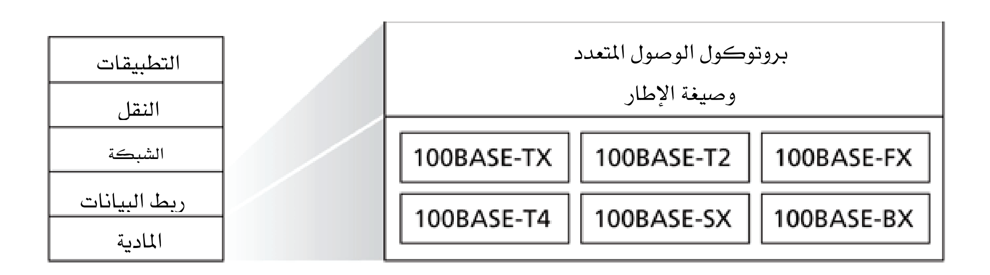
الشكل 5-24 محوّل طبقة ربط البيانات يربط بين ست عقد.

في منتصف التسعينيات ظهرت معايير إيثرنت بسرعة 100 ميجابت/ثانية، أي أسرع 10 مرات من المعيار السابق بسرعة 10 ميجابت/ثانية. تم الإبقاء على البروتوكول الأصلي للوصول المتعدد وصيغة إطار الإيثرنت، لكن وُصّفت سرعات أعلى للطبقة المادية للأسلاك النحاسية المجدولة (100BASE-T) والألياف الضوئية

(100BASE-FX, 100BASE-SX, 100BASE-BX). يبين الشكل 5-25 تلك المعايير المختلفة وبروتوكول الإيثرنت المشترك للوصول المتعدد وصيغة الإطار. يلاحَظ أن الإيثرنت بسرعة 100 ميجابت/ثانية محدودة بمسافة 100 متر فقط على زوج أسلاك النحاس المجدولة، وعدّة كيلومترات على الألياف الضوئية، مما يسمح بالتوصيل ما بين محوّلات الإيثرنت في بنايات مختلفة.

تعتبر إيثرنت الجيجابت امتداداً طبيعياً لمعايير الإيثرنت الناجحة جداً بسرعة 10 ميجابت/ثانية و100 ميجابت/ثانية. توفر إيثرنت الجيجابت معدل إرسال للبيانات قدره 1000 ميجابت/ثانية، وتحافظ على توافق كامل مع القاعدة العريضة من معدات شبكات الإيثرنت المستخدمة حالياً. يتسم معيار إيثرنت الجيجابت والمعروف بـ IEEE 802.3z بما يلي:

- يستخدم صيغة إطار الإيثرنت القياسي (الشكل 5-22) ويتوافق تراجعياً مع 10BASE-T و100BASE-T، مما يُسهّل تكامل أنظمة إيثرنت الجيجابت مع أجهزة الإيثرنت المستخدمة حالياً.
- يسمح بوصلات نقطة إلى نقطة بالإضافة إلى قنوات الإذاعة المشتركة. تستخدم وصلات نقطة إلى نقطة محوّلات (switches) بينما تستخدم قنوات الإذاعة مجمِّعات (hubs)، كما تقدّم وصفه. في مفردات إيثرنت الجيجابت يطلق على المجمِّعات موزّعات بمخازن مؤقتة (buffered distributors).



الشكل 5-25 معايير الإيثرنت 100 ميجابت/ثانية: طبقة ربط بيانات مشتركة، وطبقات مادية مختلفة.

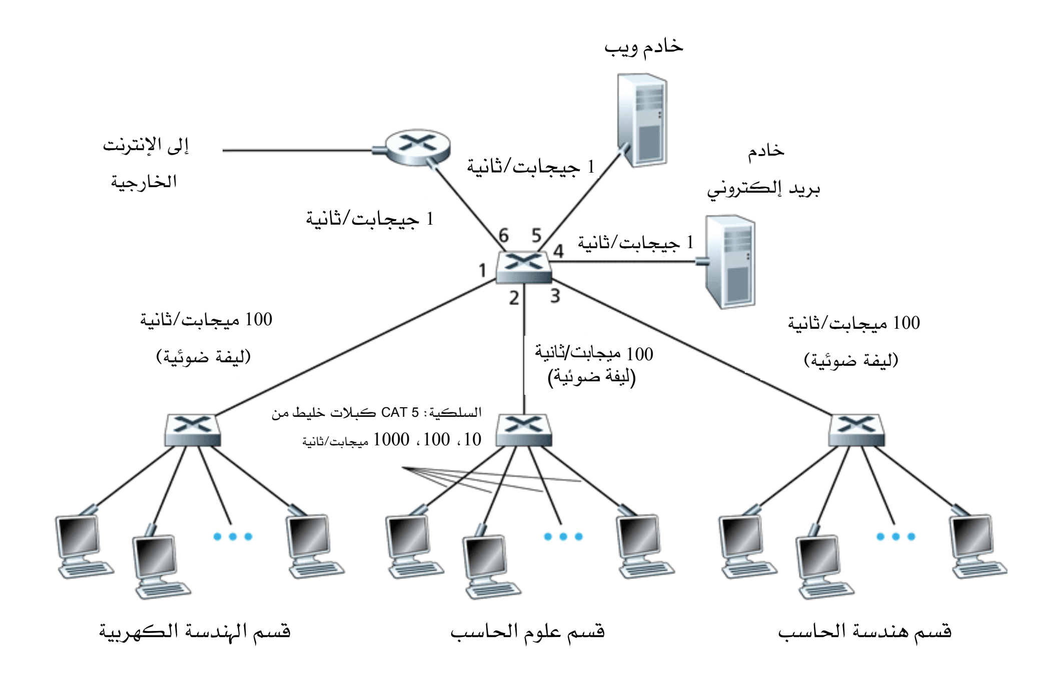
- يُستخدم بروتوكول CSMA/CD لقنوات الإذاعة المشتركة. ولتحقيق كفاءة مقبولة يجب الحد من المسافة القصوى بين العقد بشكل كبير.
- يُسمح باتصال مزدوج بالكامل (full-duplex) بمعدّل إرسال 1000 ميجابت/ثانية في كلا الاتجاهين لقنوات نقطة إلى نقطة.

في البداية كانت إيثرنت الجيجابت تتطلّب استخدام الألياف الضوئية كوسط مادي، أما الآن فيمكن استخدامها على أسلاك نحاس مجدولة من الفئة الخامسة (UTP 5). في صيف عام 2006 تم اعتماد معيار إيثرنت 10 جيجابت/ثانية (10GBASE-T)، مما يفتح المجال لشبكات إيثرنت بسعات أكبر في المُستقبل القريب.

لنختم مناقشتنا عن تقنيات الإيثرنت بطرح سؤال ربما يكون قد بدأ يلح عليك. في أيام طبوغرافية الناقل المحوري وطبوغرافية النجمة المبنية على استخدام مجمِّع، كانت الإيثرنت تمثل بوضوح وصلة إذاعة (كما عرّفناها في الجزء 5-3)، حيث تصطدم الإطارات عندما تقوم العقد بالإرسال في نفس الوقت. للتعامل مع تلك الاصطدامات تضمّن معيار الإيثرنت بروتوكول CSMA/CD، والذي يعتبر فعّالاً بصورة خاصة على شبكة بيانات محلية بوصلة إذاعة تمتدّ عبر نصف قطر صغير. لكن إذا كان الاستعمال السائد للإيثرنت اليوم يعتمد طبوغرافية نجمية مبنية على استخدام محوّل (switch)، ويطبق أسلوب "خزن ومرر" لتحويل الرزم، ألا زلنا حقاً بحاجة لبروتوكول الإيثرنت للوصول المتعدد؟ كما سنرى في الجزء 5-6 يُنسّق المحوّل إرساله بحيث لا يرسل أبداً أكثر من إطار واحد على نفس الواجهة في أي وقت. وعلاوة على ذلك فإن معظم المحوّلات الحديثة تعمل بطريقة الازدواج الكامل (full-duplex)، ومن ثم يمكن تبادل الإطارات بين المحوّل والعقدة في نفس الوقت بدون حدوث تداخل. وبمعنى آخر لا توجد اصطدامات في شبكة إيثرنت محلية مبنية على محوّل، ومن ثم فليست هناك حاجة لبروتوكول الوصول المتعدد!

كما رأينا تختلف إيثرنت اليوم جداً عن الإيثرنت الأصلية التي اخترعها ميتكالف وبوجز منذ أكثر من ثلاثين عاماً. فقد زادت سرعتها على ثلاث مراحل،

وتُنقل إطاراتها الآن على تشكيلة من أوساط النقل المادية، كما انتشرت شبكات الإيثرنت التي تستخدم محوّلات، والآن حتى بروتوكول الماك لم يعد ضرورياً في أغلب الأحيان! هل كل ذلك ما يزال إيثرنت؟ الجواب بالطبع "نعم، من حيث التعريف". من الجدير بالملاحظة أنه رغم كل هذه التغييرات، فإن ثمة شيئاً واحداً بقي بدون تغيير على مدى أكثر من ثلاثين عاماً: ألا وهو صيغة إطار الإيثرنت (قد تكون تلك هي العامل المشترك الوحيد في الواقع بين معايير الإيثرنت المختلفة).



الشكل 5-26 شبكة مؤسّسة تتضمن مجموعة من المجمِّعات، ومحوّلات الإيثرنت، وموجّه.

6-5 محوّلات طبقة ربط البيانات

كما هو مبين في الشكل 5-26 تستخدم شبكات الإيثرنت المحلية الحديثة طبوغرافية نجمية، حيث توصّل كل عقدة بمحوّل مركزي (central switch). حتى الآن كان الأمر مبهماً فيما يتعلّق بماهية ذلك المحوّل: ماذا يفعل وكيف يعمل؟ يتلخص دور المحوّل في استلام إطارات طبقة ربط البيانات من الوصلات القادمة إليه وتوصيلها إلى الوصلات الخارجة منه، وسندرس وظيفة التوجيه تلك بالتفصيل بعد قليل. يعتبر المحوّل نفسه شفافاً (transparent) (أي كأنه غير موجود) بالنسبة للعقد، بمعنى أن عقدة الإرسال تعنون الإطار إلى عقدة الاستقبال (وليس إلى المحوّل) وترسِل الإطار إلى الشبكة المحلية، وهي لا تدري أن محوّلاً سيستلم الإطار ويوجّهه إلى العقدة الأخرى. بشكلٍ مؤقت قد يتجاوز معدّل وصول الإطارات إلى أيٍّ من الواجهات الخارجة من المحوّل سعة الإرسال لوصلة تلك الواجهة. للتعامل مع هذه المشكلة تتضمن واجهات المحوّل الخارجة مخازن مؤقتة (buffers)، تقريباً بنفس الطريقة التي تستخدم بها واجهات الموجّه الخارجة المخازن المؤقتة لتخزين وحدات بيانات طبقة الشبكة. دعنا الآن نلقي نظرة متفحصة أكثر على طريقة عمل المحوّلات.

1-6-5 الترشيح والتمرير (Filtering and Forwarding)

الترشيح (filtering) هو وظيفة المحوّل التي تحدد ما إذا كان الإطار سيتم إرساله إلى واجهةٍ ما، أو أنه ببساطة سيتم إسقاطه. أما التمرير (forwarding) فهو وظيفة المحوّل التي تحدد الواجهات التي ينبغي توجيه إطار إليها، وبعد ذلك نقل الإطار إلى تلك الواجهات. يتم تنفيذ عمليتي الترشيح والتمرير عن طريق جدول المحوّل. يحتوي جدول المحوّل على مُدخلات لبعض العقد على الشبكة المحلية، ولكن ليس بالضرورة كلها. يحتوي كل مُدخل في جدول المحوّل على: (1) عنوان ماك للعقدة، و(2) واجهة المحوّل التي تقود نحو العقدة و(3) الوقت الذي تم فيه إدراج المُدخل الخاص بالعقدة في الجدول. يبين الشكل 5-27 مثالاً لجدول على المحوّل الأعلى في الشبكة المبينة في الشكل 5-26. رغم أن هذا الوصف لتمرير

الإطارات قد يبدو مشابهاً لمناقشتنا لتوجيه وحدات البيانات في الفصل الرابع، فإننا سنكتشف بعد قليل وجود اختلافات مهمة. أحد تلك الاختلافات هو أن المحوّلات توجّه الرزم بناءً على عناوين الماك وليس على عناوين IP. سنرى أيضاً أن جدول المحوّل مبني بطريقة مختلفة جداً عن جدول التوجيه على الموجّه.

العنوان	الواجهة	الوقت
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
...	...	...

الشكل 5-27 جزء من جدول المحوّل الأعلى في الشبكة المبينة في الشكل 5-26.

لفهم كيف تتم عملية الترشيح والتمرير في المحوّل، افترض أن إطاراً بعنوان الوجهة DD-DD-DD-DD-DD-DD يصل إلى المحوّل على الواجهة x. يفحص المحوّل جدوله مستخدماً عنوان الماك DD-DD-DD-DD-DD-DD كمُدخل. هناك ثلاث حالات محتملة:

- لايوجد مُدخل في الجدول لعنوان الوجهة DD-DD-DD-DD-DD-DD. في هذه الحالة يرسل المحوّل نسخاً من الإطار إلى مخزن الخرج المؤقت الخاص بكل واجهة من واجهاته ماعدا الواجهة x التي وصل منها الإطار. بمعنى آخر إذا لم يكن هناك مُدخل بالجدول يناظر عنوان الوجهة، فإن المحوّل يذيع الإطار.
- يوجد مُدخل في الجدول يربط عنوان الوجهة DD-DD-DD-DD-DD-DD بالواجهة x. في هذه الحالة الإطار قادم من قطعة من الشبكة المحلية تضم الوجهة DD-DD-DD-DD-DD-DD. وعليه فلا حاجة لتوجيه الإطار إلى أيٍّ من الواجهات الأخرى، ومن ثم يقوم المحوّل بوظيفة الترشيح وذلك بإهمال الإطار.
- يوجد مُدخل في الجدول يربط عنوان الوجهة DD-DD-DD-DD-DD-DD بواجهة y مختلفة عن x. في هذه الحالة يلزم توجيه الإطار إلى قطعة الشبكة

المحلية الموصّلة بالواجهة y. يقوم المحوّل بوظيفة التمرير بوضع الإطار في مخزن الخرج المؤقت الخاص بالواجهة y.

دعنا نطبق هذه القواعد على المحوّل في أعلى الشكل 5-26 وجدول المحوّل الموجود عليه والمبين في الشكل 5-27. افترض أن إطاراً بعنوان الوجهة 62-FE-F7-11-89-A3 يصل إلى المحوّل من الواجهة 1. يفحص المحوّل جدوله ليجد أن وجهة الإطار تقع على قطعة الشبكة المحلية الموصلة بالواجهة 1 (أي قسم الهندسة الكهربية). هذا يعني أن الإطار كان قد أذيع على قطعة الشبكة المحلية التي تتضمن الوجهة ولذلك وصل إلى الواجهة 1 على المحوّل والتي تقع أيضاً على تلك القطعة. يقوم المحوّل بوظيفة الترشيح وذلك بإهمال هذا الإطار. افترض الآن أن إطاراً آخر بنفس عنوان الوجهة السابق يصل من واجهة 2. يفحص المحوّل جدوله ثانية فيجد أن الوجهة تتبع الواجهة 1، ومن ثم يرسل ذلك الإطار إلى مخزن الخرج المؤقت الذي يسبق الواجهة 1. يتضح من هذا المثال أنه طالما كان جدول المحوّل كاملاً ودقيقاً، فإن المحوّل يرسل بالإطارات نحو وجهتها المقصودة بدون اللجوء لإذاعة أيٍّ منها.

بهذا المعنى يعتبر المحوّل "أذكى" من المجمِّع. لكن كيف يتم تهيئة جدول المحوّل هذا في المقام الأول؟ هل هناك بروتوكولات في طبقة ربط البيانات تناظر بروتوكولات التوجيه في طبقة الشبكة؟ أم أنه يتعين على مشرف الشبكة القيام بتهيئة جداول المحوّلات يدوياً بنفسه؟

5-6-2 التعلم الذاتي

تتوافر للمحوّلات خاصية رائعة (خاصةً من منظور مشرف الشبكة المُجهَد!)، حيث يمكنها إنشاء وتحديث جداولها آلياً وذاتياً وبطريقة ديناميكية - بدون أي تدخل من مشرف الشبكة أو من بروتوكول خاص بالتهيئة. وبمعنى آخر، للمحوّلات قدرة ذاتية على التعلّم. ويتحقق ذلك كالتّالي:

1. يكون جدول المحوّل فارغاً في البداية.

2. لكل إطار قادم يتم استلامه على واجهة، يُخزّن المحوّل في جدوله: (1) عنوان الماك الموجود في حقل عنوان المصدر، (2) الواجهة التي وصل منها الإطار، (3) الوقت الحالي. بهذه الطريقة يسجّل المحوّل في جدوله قطعة الشبكة المحلية التي تقع عليها عقدة إرسال كل إطار يصله. إذا كانت كل عقدة في الشبكة المحلية سترسل في النهاية إطاراً، ففي النهاية سيتم تسجيل موقع كل عقدة في الجدول.
3. يحذف المحوّل عنواناً من الجدول إذا لم تصل إطارات بذلك العنوان كعنوان مصدر خلال فترة زمنية محددة (تسمى فترة العمر). بهذه الطريقة إذا تم استبدال حاسب شخصي على الشبكة بحاسب شخصي آخر (له موائم مختلف ومن ثم عنوان ماك مختلف)، فإن عنوان الماك للحاسب الأول سيتم حذفه في النهاية من جدول المحوّل.

دعنا نطبق خاصية التعلّم الذاتي للمحوّل الموجود في أعلى الشكل 5-26 وجدول المحوّل عليه في الشكل 5-27. افترض أنه في تمام الساعة 9:39 وصل إطار بعنوان المصدر 01-12-23-34-45-56 من الواجهة 2. افترض أن هذا العنوان ليس مدرجاً في جدول المحوّل. ومن ثم يضيف المحوّل مُدخلاً جديداً إلى جدوله، كما هو مبين في الشكل 5-28.

العنوان	الواجهة	الوقت
01-12-23-34-45-56	2	9:39
62-FE-F7-11-89-A3	1	9:32
7C-BA-B2-B4-91-10	3	9:36
...	...	...

الشكل 5-28 المحوّل يتعلّم موقع الموائم بعنوان الماك 01-12-23-34-45-56.

لنواصل مسيرتنا مع نفس المثال، افترض أن فترة العمر على هذا المحوّل هي 60 دقيقة، ولم تصل إلى المحوّل أي إطارات لها عنوان المصدر 62-FE-F7-11-89-A3 بين الساعة 9:32 والساعة 10:32. وبناءً على ذلك فبحلول الساعة 10:32 سيحذف المحوّل هذا العنوان من جدوله.

تُعدُّ المحوّلات أدوات من نوع "وصّل وشغّل" (plug-and-play)، بمعنى أنها لا تتطلب أي تدخل من مشرف الشبكة أو مستخدمها. فأي مشرف للشبكة يريد تركيب محوّل، ليس عليه إلا توصيل قطع الشبكة المحلية إلى واجهات المحوّل. لا يحتاج المشرف للقيام بتهيئة المحوّل عند تركيبه ولا عند إزالة مضيف من على إحدى قطع الشبكة المحلية. جدير بالذكر أيضاً أن المحوّلات تعمل بازدواج كامل (full-duplex)، بمعنى أنه على أي وصلة تربط عقدة بالمحوّل، يمكن لكلٍّ من العقدة والمحوّل أن يرسلا في نفس الوقت بدون حدوث أي اصطدامات.

5-6-3 خصائص التحويل في طبقة ربط البيانات

بعد أن انتهينا من وصف أساسيات تشغيل محوّلات طبقة ربط البيانات، دعنا الآن نتناول السمات والخصائص المميزة لتلك المحوّلات. بالرجوع إلى شبكة البيانات المحلية المبينة في الشكل 5-24، يمكننا التعرف على عدة مزايا لاستعمال المحوّلات بدلاً من وصلات الإذاعة التي تستخدم ناقلات (buses)، أو طبوغرافية نجمية مبنية على مجمِّع:

- تجنّب حدوث اصطدامات: في شبكة بيانات محلية مبنية باستخدام محوّلات (وبدون مجمِّعات)، لا يُفقد حيز ترددي (سعة إرسال) بسبب الاصطدامات! تقوم المحوّلات بتخزين الإطارات في المخزن المؤقت، ولا ترسل في أي وقت أبداً أكثر من إطار واحد إلى أي قطعة من قطع الشبكة. كما هو الحال مع الموجّه في طبقة الشبكة، الطاقة الإنتاجية الكلية القصوى لمحوّل هي مجموع معدلات الإرسال على كل واجهات المحوّل. وهكذا توفر المحوّلات تحسناً كبيراً في أداء شبكات البيانات المحلية مقارنةً بوصلات الإذاعة.

- إمكانية استخدام وصلات متباينة: نظراً لأن المحوّل يعزل كل وصلة من وصلاته عن الأخرى، يمكن للوصلات المختلفة في شبكة محلية أن تعمل بسرعات مختلفة وتستخدم أوساط نقل مادية مختلفة. فمثلاً على الشبكة المبينة في الشكل 5-24، يمكن توصيل المضيف A باستخدام أسلاك نحاسية بمعيار 10BASE-T بمعدّل إرسال 10 ميجابت/ثانية، في حين يوصل المضيف B بواسطة ليفة ضوئية بمعيار 100BASE-FX ومعدّل إرسال 100 ميجابت/ثانية، وC عبر أسلاك نحاسية بمعيار 1000BASE-T بمعدّل إرسال 1 جيجابت/ثانية. وهكذا تُعدُّ المحوّلات طريقةً مثاليةً للجمع ما بين الأجهزة القديمة والأجهزة الحديثة على نفس الشبكة.
- إدارة الشبكة: بالإضافة إلى تحسين أمن الشبكة (انظر المادة الجانبية بعنوان "نبذة عن الأمن")، تسهم المحوّلات كذلك في التخفيف من أعباء إدارة الشبكة. فعلى سبيل المثال إذا تعطل موائم على الشبكة وصار يرسل إطارات إيثرنت بشكلٍ مستمر (يطلق عليه عندئذٍ الموائم "الثرثار")، يمكن للمحوّل أن يكتشف هذه المشكلة تلقائياً ويفصل الموائم المعطوب عن الشبكة. بهذه الميزة لن يحتاج مشرف الشبكة لأن يغادر فراشه ليلاً ويقود سيارته إلى محل عمله لحل المشكلة. أيضاً إذا انقطع كبل فسيؤدي ذلك فقط إلى فصل تلك العقدة التي كانت تستخدم الكبل المقطوع للوصول إلى المحوّل. في أيام الكبل المحوري كان الكثير من مشرفي الشبكة يقضون الساعات لتتبع الكبل للعثور على مكان انقطاعه الذي عطّل الشبكة بكاملها. كما سنرى في الفصل التاسع (إدارة الشبكات)، تجمع المحوّلات أيضاً إحصائيات عن استغلال الحيز الترددي، ومعدّلات الاصطدام، وأنواع حركة البيانات. وتقدّم هذه المعلومات إلى مشرف الشبكة. يمكن استخدام تلك المعلومات لتحرّي الأعطال، وحل المشاكل، وللتخطيط من أجل تطوير الشبكة المحلية في المُستقبل.

نبذة عن الأمن (Focus on Security)

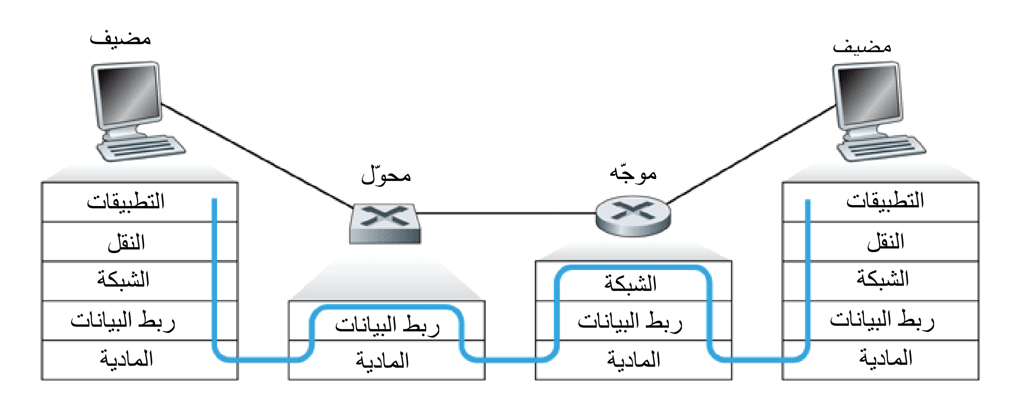
التقاط الرزم من الشبكات المحوّلة بتسميم المحوّل

عندما توصّل عقدة إلى محوّل فإنها تستلم عادةً الإطارات التي ترسل إليها على وجه التحديد فقط. على سبيل المثال خذ في الاعتبار شبكة البيانات المحلية في الشكل 5-24. عندما ترسل العقدة A إطاراً إلى العقدة B، ويكون هناك مُدخل للعقدة B في جدول المحوّل، فسيقوم المحوّل بإرسال ذلك الإطار فقط إلى العقدة B. إذا صادف وكانت العقدة C تشغّل برنامجاً لإلتقاط الرزم، فلن يتمكن ذلك البرنامج من التقاط ذلك الإطار من A إلى B. وهكذا ففي بيئة شبكة محلية محوّلة (LAN switched) (في مقابل بيئة شبكة محلية ذات وصلة إذاعة كـ 802.11 أو مبنية على استخدام مجمِّع)، يُعتبر التقاط الإطارات من قِبَل مهاجم أمراً أكثر صعوبة. ومع ذلك، فنظراً لأن المحوّل سيذيع الإطارات التي تكون عناوين الوجهة لها غير موجودة في جدول المحوّل، فلا يزال بوسع لاقط الرزم على C التقاط بعض الإطارات التي ليست معنونة إلى C بالتحديد. وعلاوة على ذلك سيكون بوسع لاقط الرزم التقاط كل إطارات الإيثرنت المذاعة التي تحمل العنوان المخصص للإذاعة (FF-FF-FF-FF-FF-FF) كعنوان الوجهة. من أنواع الهجوم المشهورة ضد المحوّلات هجوم يعرف بتسميم المحوّل (switch poisoning). في هذا الهجوم يتم إرسال أطنان من الرزم إلى المحوّل تحمل العديد من عناوين الماك المختلفة والمزيفة للمصدر. يؤدي ذلك إلى ملء جدول المحوّل بمُدخلات مزيفة، بحيث لا يبقى ثمة مكان لعناوين الماك التي تستخدمها العقد الشرعية. يؤدي ذلك بالمحوّل إلى إذاعة أكثر الإطارات، وعندئذٍ يمكن التقاطها بواسطة لاقط الرزم [Skoudis 2006]. ونظراً لأن هذا الهجوم يُعد معقّداً حتى بالنسبة لمهاجم محنّك، فإن المحوّلات تعتبر أقل عرضة لالتقاط الرزم بدرجة كبيرة مقارنةً بالشبكات المحلية اللاسلكية وتلك المبنية على استخدام مجمِّعات.

5-6-4 المحوّلات في مقابل الموجّهات

كما رأينا في الفصل الرابع فإن الموجّهات هي محوّلات رزم تعمل بطريقة "خزّن ومرر" لتوجيه الرزم على أساس عنوان طبقة الشبكة الذي تحمله كل رزمة. رغم أن المحوّل يعتبر أيضاً محوّل رزم من نوع "خزّن ومرر" فإنه يختلف جوهرياً عن الموجّه، حيث إنه يوجّه الرزم مستخدماً عناوين ماك. وباختصار: الموجّه هو محوّل رزم في طبقة -3، أما المحوّل فهو محوّل رزم في طبقة -2.

رغم إن المحوّلات والموجّهات أدوات مختلفة بشكلٍ جوهري، فإنه غالباً ما يتعين على مشرفي الشبكات الاختيار بينهما عند تركيب أداة تشبيك. على سبيل المثال كان بوسع المشرف على الشبكة في الشكل 5-26 أن يستخدم بسهولة موجّهاً بدلاً من محوّل لتوصيل الشبكات المحلية للأقسام، والخادمات، وموجّه بوّابة الإنترنت. في الحقيقة سيسمح الموجّه بالاتصالات بين الأقسام بدون اصطدامات. ولمّا كانت كلٌّ من المحوّلات والموجّهات مرشّحة للاستخدام كأدوات تشبيك، يجدر بنا معرفة مزايا وعيوب كلٍّ منها.



الشكل 5-29 معالجة الرزم في المحوّلات، والموجّهات، والمضيفات.

لنتناول مزايا وعيوب المحوّلات أولاً. كما ذكرنا أعلاه فإن المحوّلات أجهزة من نوع "وصّل وشغّل"، وهي خاصية يقدّرها كل مشرفي الشبكات في العالم. المحوّلات يمكنها ترشيح وتوجيه الإطارات بمعدّلات عالية نسبياً. كما يوضح الشكل 5-29 يجب على المحوّلات معالجة الإطارات فقط حتى الطبقة 2، أما الموجّهات فيجب أن تعالج وحدات البيانات حتى الطبقة 3. من ناحية أخرى ولمنع دوران الإطارات المذاعة فإنه يجب ألا تتجاوز الترتيبة الفعّالة للشبكة المحوّلة شجرة اتصال ممتدة (spanning tree). أيضاً تتطلب شبكة محوّلة كبيرة جداول كبيرة في العقد لبروتوكول تحويل العناوين ARP، كما تولد حركة مرور ومعالجة كبيرة تتعلق بهذا البروتوكول. وعلاوة على ذلك لا توفر المحوّلات أي حماية ضد عاصفة

البث الإذاعي (broadcast storm) – فإذا أخذ مضيف على الشبكة يرسل سلسلة لانهائية من إطارات الإيثرنت المذاعة، فستقوم المحوّلات بتمرير كل تلك الإطارات – مما يؤدي إلى انهيار الشبكة بالكامل.

لنتناول الآن مزايا وعيوب الموجّهات. نظراً لأن عنونة الشبكة هرمية (hierarchical) في أغلب الأحيان وليست مسطّحة (flat) كما في عنونة الماك، فإن الرزم لا يتكرر دورانها خلال الموجّهات حتى عندما تتضمن الشبكة مسارات إضافية (لاحظ أن دوران الرزم قد يحدث إذا لم يتم تهيئة جداول التوجيه بشكلٍ جيد. ولكن كما رأينا في الفصل الرابع، يستخدم بروتوكول IP حقلاً خاصاً في ترويسة وحدة البيانات للحد من دوران الرزم). وعليه فلن يتم حصر الرزم في نطاق شجرة الاتصال الممتدة، وسيمكنها استخدام أفضل مسار بين المصدر والوجهة. ونظراً لأن الموجّهات لا تعاني من قيود شجرة الاتصال الممتدة، فقد سمح ذلك ببناء إنترنت بترتيبات غنية ـ تتضمّن على سبيل المثال وصلات متعددة نشطة بين أوروبا وأمريكا الشمالية. من المزايا الأخرى للموجّهات أنها توفر حماية ببرامج الـ firewall ضد عواصف الإذاعة في الطبقة 2. في المقابل لعل العائق الأساسي لاستخدام الموجّهات هو أنها ليست أجهزة من نوع "وصّل وشغّل"، فهي تحتاج مع المضيفات الموصّلة بها إلى تهيئة عناوين IP الخاصة بها يدوياً. كما أن الموجّهات غالباً ما تستغرق وقتاً أطول لمعالجة كل رزمة مقارنةً بالمحوّلات، نظراً لأن عليها المعالجة حتى الطبقة 3. وأخيراً هناك طريقتان مختلفتان لنطق اسم الموجّه (router) باللغة الإنجليزية، إمّا "rootor" أو "rowter"، ويضيّع الناس الكثير من الوقت في الجدل حول أيّ الطريقتين أصح [Perlman 1999].

الآن وبعد أن عرفنا مزايا وعيوب كلٍّ من المحوّلات والموجّهات، متى إذن يجدر بشبكة مؤسّسة (كشبكة في حرم جامعي أو شركة) استخدام محوّلات، ومتى يستحسن أن تستخدم موجّهات؟

عادةً ما تتألف الشبكات الصغيرة التي تضم بضع مئات من المضيفات من بضع قطع (segments) من الشبكات المحلية. تكفي المحوّلات لهذه الشبكات

الصغيرة، حيث تفيد في زيادة محلية حركة مرور البيانات وتزيد الطاقة الإنتاجية الكلية دون الحاجة لأي تهيئة لعناوين IP. أما الشبكات الأكبر التي تشمل آلاف المضيفات فعادةً ما تتضمّن موجّهات ضمن الشبكة (بالإضافة إلى المحوّلات). تحقق الموجّهات عزلاً أكثر متانة لحركة المرور، وتحكماً أفضل في عواصف الإذاعة، كما تستخدم مسارات "ذكية" أكثر بين المضيفات في الشبكة.

عرفنا في هذا الجزء أنه يمكن استخدام كلٍّ من المجمِّعات، والمحوّلات والموجّهات كأدوات لتشبيك قطع الشبكات المحلية والمضيفات. يلخّص الجدول 5-1 أبرز السمات التي تميز كل أداة من أدوات التشبيك تلك.

الخاصيّة	المجمِّعات	الموجّهات	المفاتيح
عزل حركة المرور	لا	نعم	نعم
سمة "وصّل وشغّل"	نعم	لا	نعم
التوجيه الأمثل	لا	نعم	لا
توفير طرق مختصرة	نعم	لا	نعم

الجدول 5-1 مقارنة بين الخصائص النمطية لأدوات التشبيك الشهيرة.

5-7 بروتوكول نقطة إلى نقطة (PPP)

تركزت أغلب مناقشاتنا لبروتوكولات طبقة ربط البيانات حتى الآن على بروتوكولات قنوات الإذاعة. سنتناول في هذا الجزء بروتوكولاً آخر لطبقة ربط البيانات مصمماً للتعامل مع الوصلات من نقطة إلى نقطة، وهو بروتوكول نقطة إلى نقطة (Point-to-Point Protocol (PPP)). نظراً لكون PPP هو البروتوكول المفضّل لوصلات المودم الهاتفي (dial-up links) من المضيفات السكنية، فإنه يعتبر بلا شك أحد أكثر بروتوكولات طبقة ربط البيانات انتشاراً اليوم. البروتوكول الآخر المهم لطبقة ربط البيانات والمستخدم اليوم هو بروتوكول المستوى العالي للتحكم في وصلة ربط البيانات (HDLC)؛ ويتضمن [Spragins 1991] مناقشة لبروتوكول HDLC. ستمكنّنا مناقشتنا هنا لبروتوكول PPP الأسهل من

استكشاف العديد من السمات الهامة لبروتوكولات طبقة ربط البيانات من نوع نقطة إلى نقطة.

كما يدل الاسم، بروتوكول نقطة إلى نقطة [RFC 1661; RFC 2153] هو بروتوكول لطبقة ربط البيانات يعمل على وصلة من نقطة إلى نقطة - أي وصلة تربط مباشرةً بين عقدتين، تقع كل عقدة على طرف من طرفي الوصلة. يمكن أن تكون وصلة النقطة إلى نقطة التي يعمل عليها بروتوكول PPP خطاً هاتفياً تسلسلياً بمودم (على سبيل المثال، وصلة مودم بسرعة 56 كيلوبت/ثانية، أو وصلة SONET/SDH، أو وصلة X.25 ، أو دائرة ISDN). وكما ذكرنا أعلاه أصبح PPP البروتوكول المفضّل لتوصيل المستخدمين السكنيين في منازلهم إلى موفري خدمة الإنترنت لهم على وصلات مودم هاتفية. قبل الخوض في تفاصيل بروتوكول PPP، من المفيد استعراض المتطلبات الأصلية التي حددها فريق عمل هندسة الإنترنت (IETF) لتصاميم PPP [RFC 1547]:

- تأطير الرزم: ينبغي أن يكون بوسع المُرسِل ببروتوكول PPP على وصلة ربط البيانات أخذ رزمة من مستوى الشبكة وتغليفها ضمن إطار PPP لطبقة ربط البيانات بحيث يمكن لمُستقبِل الإطار تحديد بداية ونهاية كلٍّ من إطار طبقة ربط البيانات ورزمة طبقة الشبكة المتضمنة في الإطار.
- الشفافية: لا ينبغي أن يفرض بروتوكول PPP أي قيود على البتات التي توضع في رزمة طبقة الشبكة (سواءً الترويسات أو البيانات). وعليه فلا يمكن لبروتوكول PPP مثلاً منع استعمال تسلسل معين من البتات في رزمة طبقة الشبكة. سنعود إلى هذه القضية بعد قليل أثناء مناقشتنا لموضوع حشو البايتات.
- دعم عدة بروتوكولات لطبقة الشبكة: يجب أن يكون بروتوكول PPP قادراً على دعم العديد من بروتوكولات طبقة الشبكة (مثل IP وDECnet) التي تستخدم نفس الوصلة المادية في نفس الوقت. تماماً كما يحتاج بروتوكول IP للقدرة على تجميع البيانات (multiplex) من بروتوكولات مختلفة بطبقة نقل البيانات (مثل TCP وUDP) على توصيلة واحدة من طرف

إلى طرف، يحتاج بروتوكول PPP أن يكون لديه القدرة على تجميع البيانات من عدة بروتوكولات طبقة شبكة مختلفة على وصلة واحدة من نقطة إلى نقطة. يعني هذا المتطلب أنه في الحد الأدنى ينبغي أن يتضمن بروتوكول PPP حقلاً أو آلية أخرى مماثلة لتحديد نوع بروتوكول طبقة الشبكة المستخدم، بحيث يتسنى لجانب الاستقبال من بروتوكول PPP توزيع (demultiplex) الإطار المستلَم إلى البروتوكول المناظر في طبقة الشبكة.

- دعم أنواع متعددة من الوصلات: بالإضافة إلى قدرته على التعامل مع عدة بروتوكولات في المستوى الأعلى، يجب أن يكون بوسع بروتوكول PPP العمل على تشكيلة كبيرة من الأنواع المختلفة من الوصلات، بما في ذلك الوصلات التسلسلية (التي ترسل البيانات في اتجاه معين على شكل بتات الواحد تلو الآخر) أو المتوازية (التي ترسل عدة بتات في نفس الوقت على التوازي)، وكذلك الوصلات المتزامنة (التي ترسل إشارة ساعة توقيت مع بتات البيانات) أو غير المتزامنة، وكذلك الوصلات منخفضة أو عالية السرعة، والوصلات الكهربية أو الضوئية.
- اكتشاف الأخطاء: يجب أن يكون لمُستقبِل بروتوكول PPP القدرة على اكتشاف أخطاء البتات في الإطار المستلَم.
- حيوية التوصيلة: يجب أن يكون لبروتوكول PPP القدرة على اكتشاف الأعطال على مستوى الوصلة (كعدم القدرة على نقل البيانات من جانب الإرسال إلى جانب الاستقبال من الوصلة) وإرسال إشارة بذلك إلى طبقة الشبكة.
- مفاوضات عنوان طبقة الشبكة: ينبغي أن يوفر بروتوكول PPP آلية لطبقات الشبكة المتصلة (على سبيل المثال IP) لمعرفة وتهيئة عنوان طبقة الشبكة لبعضها البعض.
- البساطة: كان على بروتوكول PPP تحقيق عدد من المتطلبات الأخرى بالإضافة لتلك المدرجة أعلاه، وكان على قمة كل تلك المتطلبات قبل كل شيء البساطة. تنص الوثيقة RFC 1547 على أن "الشعار الذي ينبغي أن يميّز

بروتوكول نقطة إلى نقطة PPP يجب أن يكون البساطة". وياله من مطلب صعب المنال في الواقع - إذا ما أخذنا في الاعتبار القائمة الطويلة من المتطلبات الأخرى لتصميم بروتوكول PPP. ظهر أكثر من خمسين من طلبات التعليقات (RFCs) حتى الآن لتعريف الجوانب المختلفة لهذا البروتوكول "البسيط"!

رغم أن قائمة المتطلبات التي وُضعت لتصميم بروتوكول PPP قد تبدو طويلة، إلا أن الوضع كان يمكن أن يكون أسوأ من ذلك! فمواصفات التصميم للبروتوكول نصت أيضاً على وظائف لم يكن مطلوباً من بروتوكول PPP أن يحققها، مثل:

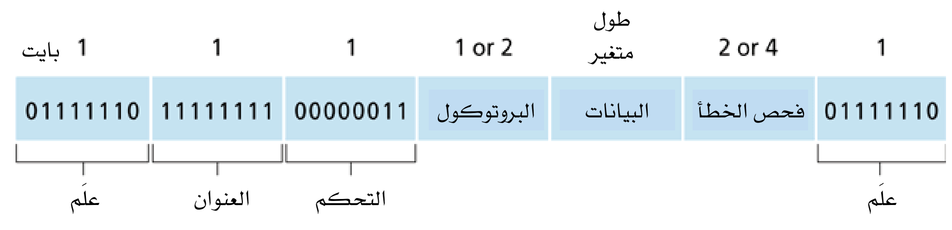
- تصحيح أخطاء البيانات: فبروتوكول PPP مطلوب منه اكتشاف أخطاء البتات ولكن ليس مطلوباً منه تصحيحها.
- ضبط التدفق: يُتوقع من مُستقبِل بروتوكول PPP أن يكون قادراً على استلام الإطارات عند إرسالها بمعدل الإرسال الكامل للطبقة المادية التحتية. إذا كانت طبقة أعلى لا تستطيع استلام الرزم بمعدل الإرسال هذا، فإن الطبقة الأعلى تتحمل مسؤولية إسقاط (إهمال) بعض الرزم أو "خنق" المُرسِل في الطبقة الأعلى. بمعنى أنه بدلاً من جعل مُرسِل PPP يخنق معدل إرساله بنفسه، يتحمل بروتوكول المستوى الأعلى مسؤولية خنق المعدل الذي يسلّم به مُرسِل ذلك البروتوكول الرزم إلى بروتوكول PPP لتوصيلها.
- تسلسل الإطارات: ليس مطلوباً من بروتوكول PPP توصيل الإطارات إلى مُستقبِل الوصلة بنفس الترتيب الذي أُرسِلَت به. من الجدير بالملاحظة أنه في حين تلائم تلك المرونة نموذج الخدمة لبروتوكول IP (والذي يسمح بتوصيل رزم IP من طرف إلى طرف بأي ترتيب)، فإن بروتوكولات أخرى لطبقة الشبكة والتي تعمل فوق بروتوكول PPP تتطلّب توصيل الرزم من طرف إلى طرف بالترتيب.
- الوصلات متعددة النقاط: المطلوب من بروتوكول PPP العمل فقط على الوصلات التي عليها مُرسِل واحد ومُستقبِل واحد. يمكن لبروتوكولات

أخرى لطبقة ربط البيانات (مثل بروتوكول HDLC) التعامل مع عدة مُستقبِلين على وصلة (على سبيل المثال سيناريو شبيه بالإيثرنت) .

بعد أن تناولنا أهداف التصميم لبروتوكول PPP، دعنا نرى كيف استطاع تصميم هذا البروتوكول تحقيق هذه الأهداف.

5-7-1 تأطير البيانات في بروتوكول PPP

يبين الشكل 5-30 إطار بيانات PPP والذي يستخدم أسلوب تأطير مماثل لذلك المستخدم في بروتوكول HDLC [RFC 1662]. يتضمن إطار PPP الحقول التالية:



الشكل 5-30 صيغة إطار البيانات في بروتوكول PPP.

- حقل العَلَم (flag) : يبدأ كل إطار PPP وينتهي بحقل خاص طوله بايت واحد قيمته 01111110.
- حقل العنوان (address): القيمة الوحيدة المحتملة لهذا الحقل هي 11111111.
- حقل التحكم (control): القيمة المحتملة الوحيدة لهذا الحقل هي 00000011. لمّا كان كلٌّ من حقلي التحكم والعنوان يأخذ قيمة واحدة (ثابتة) فقط، فقد تتساءل: لماذا تُعرّف تلك الحقول في المقام الأول. تنص مواصفات بروتوكول PPP [RFC 1662] على أنه قد يتم تعريف قيم أخرى في وقت لاحق، رغم أنه لم يتم شيء من ذلك حتى الآن. نظراً لأن هذه الحقول تأخذ قيماً ثابتة، فإن بروتوكول PPP يسمح للمُرسِل ببساطة بعدم إرسال بايتات العنوان والتحكم، ومن ثم يوفّر بايتين اثنين من العبء الإضافي في كل إطار PPP.

- حقل البروتوكول (protocol): يُخبر هذا الحقل مُستقبِل PPP ببروتوكول الطبقة الأعلى الذي تنتمي له البيانات المغلّفة التي تم استلامها (أي محتويات حقل المعلومات في إطار PPP). عند استلام إطار PPP يقوم مُستقبِل PPP بفحص الإطار للتأكد من صحته ثم يمرر البيانات المغلّفة إلى البروتوكول المناظر. يعرّف كلٌّ من RFC 1700 وRFC 3232 أرقام البروتوكولات التي يستخدمها بروتوكول PPP. إننا نهتم ببروتوكول IP في طبقة الشبكة (حيث تمثل البيانات المغلّفة في إطار PPP رزمة بيانات IP). يناظر بروتوكول IP القيمة 21 (بالصيغة الست عشرية) لحقل البروتوكول في إطار PPP. من بروتوكولات طبقة الشبكة الأخرى بروتوكول AppleTalk وبروتوكول DECnet وتُمثّل بالقيم 29 و27 على الترتيب.
- حقل المعلومات (information): يحتوي هذا الحقل على الرزمة المغلّفة (البيانات) التي يرسلها بروتوكول طبقة أعلى (مثلاً بروتوكول IP) على وصلة PPP. يبلغ الطول الأقصى المعتاد لحقل المعلومات 1500 بايت، مع أنه يمكن تغيير تلك القيمة عند تهيئة الوصلة في البداية كما سنبين لاحقاً.
- حقل المجموع التدقيقي (checksum): يُستخدم هذا الحقل لاكتشاف أخطاء البتات في الإطارات المُرسَلة. تُستعمل شفرة تدقيق إضافية دورية تبعاً لمعيار HDLC بطول بايتين أو 4 بايتات.

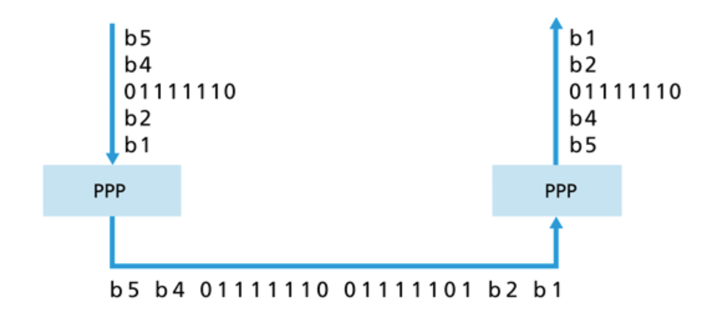
حشو البايتات (Byte Stuffing)

قبل أن نختم مناقشتنا لإطارات PPP، دعنا نتناول مشكلة تظهر عندما يستخدم بروتوكولٌ ما مسلسل بتات معين في حقل العَلَم لتحديد بداية أو نهاية الإطار. ماذا يحدث لو تكرر مسلسل بتات العَلَم في مكان آخر داخل الرزمة؟ على سبيل المثال ماذا يحدث لو ظهرت قيمة حقل العَلَم 01111110 في حقل المعلومات داخل الإطار؟ هل يتصوّر المستلم أنه اكتشف نهاية إطار PPP بشكلٍ خاطئ؟

يكمن أحد الطرق لحل هذه المشكلة في أن يمنع بروتوكول PPP بروتوكول الطبقة الأعلى من إرسال بيانات تحتوي على مسلسل بتات حقل العَلَم. غير أن متطلب

الشفافية في بروتوكول PPP والذي ذكرناه آنفاً يحول دون استخدام هذا الحل. هناك حل بديل، وهو المستخدم في بروتوكول PPP والعديد من البروتوكولات الأخرى، ويتلخص في استخدام التقنية المعروفة بحشو البايتات (byte stuffing).

يُعرّف PPP بايت تحكم خاص للهروب قيمته 01111101. إذا حدث وظهرت قيمة حقل العَلَم 01111110 في أي مكان في الإطار، باستثناء حقل العَلَم، يُدخل بروتوكول PPP قبل ذلك البايت بايت تحكم الهروب. بمعنى أنه "يحشو" (يضيف) بايت تحكم الهروب في سلسلة البيانات المُرسَلة، قبل 01111110، للإشارة إلى أن البايت التالي (01111110) ليس قيمة العَلَم ولكن في الحقيقة يمثّل بيانات فعلية. أي مُستقبِل يرى 01111110 مسبوقاً بـ 01111101 سيقوم بالطبع بإزالة بايت تحكم الهروب التي قام المُرسِل بحشوه وذلك لاستعادة سلسلة البيانات الأصلية. بنفس الطريقة، إذا ظهر بايت تحكم الهروب نفسه ضمن البيانات الفعلية، يجب أيضاً أن تُسبق ببايت تحكم هروب آخر يتم حشوه. وهكذا فعندما يرى المُستقبِل بايت تحكم هروب لوحده في سلسلة البيانات فإنه يعرف إن البايت تم حشوه في سلسلة البيانات. أما إذا ظهر زوج من بايتات تحكم الهروب (الواحد تلو الآخر مباشرةً) فهذا يعني أن البيانات الأصلية المُرسَلة تحتوي على بايت تحكم هروب واحد. يوضح الشكل 5-31 عملية حشو البايتات في بروتوكول PPP. (في الحقيقة يقوم PPP أيضاً بإجراء عملية "أو - الحصرية" (XOR)، بين بايت البيانات الذي يتم الهروب منه والرقم الست عشري 20، ولكن هذا تفصيلٌ آثرنا إهماله بهدف التبسيط).



الشكل 5-31 حشو البايتات في بروتوكول PPP.

نشير هنا إلى أن بروتوكول PPP يتضمن أيضاً بروتوكولاً للتحكم في الوصلة ((link control protocol (LCP)) والذي تتلخص وظيفته في تهيئة وصيانة وإغلاق وصلة PPP. تتضمن المواد الإضافية المرتبطة بهذا الكتاب على الإنترنت مناقشة عن بعض تفاصيل بروتوكول LCP.

8-5 الوصلة الافتراضية: الشبكة كطبقة ربط البيانات

لمّا كان هذا الفصل يتعلّق ببروتوكولات طبقة الوصلة، وبما أننا نقترب الآن من نهاية الفصل، دعنا نتأمل كيف تطوّر فهمنا للمصطلح "وصلة". لقد بدأنا هذا الفصل بالنظر إلى الوصلة كسلك مادي يصل ما بين مضيفين يتصلان فيما بينهما كما وضّح الشكل 5-2. في دراستنا لبروتوكولات الوصول المتعدد (الشكل 5-9)، رأينا أنه يمكن ربط عدة مضيفات ببعضها بواسطة سلك مشترك، وأن "السلك" الذي يربط المضيفات يمكن أن يكون حيز ترددات لاسلكية أو وسطاً مادياً آخر. بهذا المفهوم بدأنا ننظر إلى الوصلة بشيء من التجريد كـ "قناة"، بدلاً منها كـ "سلك". في دراستنا لشبكات الإيثرنت المحلية (الشكل 5-26) رأينا أن أوساط الربط المادية يمكن أن تكون في الحقيقة بنية نقل تحتية معقدة، ومع ذلك فعبر مراحل هذا التطور احتفظت المضيفات نفسها بالمفهوم ذاته ـ أن وسط التشبيك هو ببساطة قناة طبقة ربط بيانات تصل ما بين مضيفين أو أكثر. رأينا على سبيل المثال أن مضيفاً على الإيثرنت يمكن أن يكون غير مدرك لما إذا كان موصّلاً بالمضيفات الأخرى على الشبكة المحلية بواسطة قطعة واحدة قصيرة من شبكة محلية (الشكل 5-9) أو عبر شبكة محلية متسعة جغرافياً وتستخدم المحوّلات (الشكل 5-26).

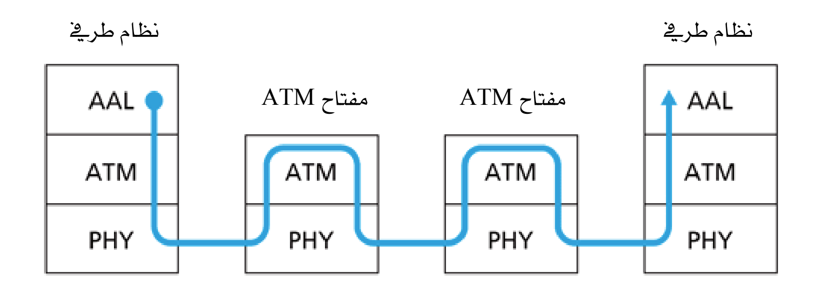
في الجزء 5-7 رأينا أن بروتوكول PPP يُستخدم غالباً عبر وصلة مودم بين مضيفين. في هذه الحالة، الوصلة التي تربط بين المضيفين هي في الحقيقة شبكة الهاتف ـ وهي شبكة اتصالات عالمية مستقلة منطقياً لها محوّلاتها، ووصلاتها، ورصات البروتوكولات الخاصة بها لنقل البيانات وإرسال إشارات التحكم (التأشير) (signaling). ولكن من وجهة نظر طبقة ربط البيانات في الإنترنت، يُنظر

إلى توصيلة المودم عبر شبكة الهاتف على أنها ببساطة سلك. بهذا المعنى فإن الإنترنت "تُجرّد" شبكة الهاتف، حيث تعتبرها بمثابة تقنية افتراضية لطبقة ربط البيانات توفر اتصالاً بين اثنين من مضيفي الإنترنت. تذكّر من مناقشتنا لمفهوم الشبكة الإضافية (overlay network) في الفصل الثاني أن الشبكة الإضافية تنظر إلى الإنترنت بنفس الطريقة كوسيلة لتوفير توصيلات بين عقد الشبكة الإضافية، بحيث تغطي (overlay) الإنترنت بنفس الطريقة التي تغطي بها الإنترنت شبكة الهاتف.

سنتناول في هذا الجزء شبكات نمط النقل غير المتزامن (ATM) وشبكات تحويل الوسمة متعدد البروتوكول (MPLS). على خلاف شبكة الهاتف بتحويل الدوائر تعتبر كلٌّ من ATM وMPLS بحكم تكوينهما شبكات تحويل رزم بدوائر افتراضية. لتلك الشبكات صيغ للإطارات وأساليب للتوجيه خاصة بها. وعليه فمن وجهة نظر تعليمية بحتة، من الملائم دراسة شبكات ATM وMPLS في سياق طبقة الشبكة أو طبقة ربط البيانات. غير أنه، من وجهة نظر الإنترنت، يمكن أن نعتبر شبكات ATM وMPLS مثل شبكة الهاتف وشبكات الإيثرنت المحوّلة، كتقنيات طبقة ربط بيانات توفر خدمة لتشبيك أجهزة IP. وعليه فسنتناول كلاً من شبكات ATM وMPLS في مناقشتنا لطبقة ربط البيانات. يمكن أيضاً استخدام شبكات ترحيل الإطارات (frame-relay) في تشبيك أجهزة IP على الرغم من أنها تمثل تقنيةً أقدم قليلاً (لكن لا تزال تُستخدم)، ولن نغطيها هنا وإنما ننصح بمراجعة الكتاب الجيد [Goralski 1999] للمزيد من التفاصيل. ستكون معالجتنا لشبكات ATM وMPLS مختصرة بالضرورة، فهناك كتب بكاملها تناولت تلك الشبكات. نوصي بمراجعة [Black 1995, Black 1997] و [Davie 2000] للمزيد من التفاصيل عن شبكات ATM وMPLS على الترتيب. سنركّز هنا بشكلٍ رئيس على الكيفية التي توفر بها تلك الشبكات خدمة تشبيك لأجهزة IP، مع أننا سنغوص أيضاً بعض الشيء في التقنيات التحتية المستخدمة.

5-8-1 شبكات نمط النقل غير المتزامن (ATM)

تم تطوير معايير شبكات نمط النقل غير المتزامن لأول مرة في منتصف الثمانينيات بهدف تصميم تقنية شبكات واحدة لنقل مواد الصوت والفيديو الفورية بالإضافة إلى النصوص، والبريد الإلكتروني، وملفات الصور. شاركت مجموعتان في تطوير معايير شبكات ATM، هما: منتدى ATM (والذي يُعرف الآن بمنتدى MFA Forum [MFA Forum 2007] والاتحاد الدولي للاتصالات [ITU 2007]. تم تحديد معيار كامل من طرف إلى طرف اشتمل على مواصفات تراوحت من واجهات التطبيقات مع شبكة ATM إلى تأطير بيانات ATM على مستوى البتات عبر مختلف الطبقات المادية بما في ذلك الألياف الضوئية، والأسلاك النحاسية، والراديو. عملياً استُخدمت شبكات ATM ضمن شبكات الهاتف وشبكات IP كتقنية لطبقة ربط البيانات مثلاً لتوصيل موجّهات IP كما بينّا سابقاً.



الشكل 5-32 طبقات شبكة ATM الثلاث. توجد الطبقة AAL فقط على حواف شبكة ATM.

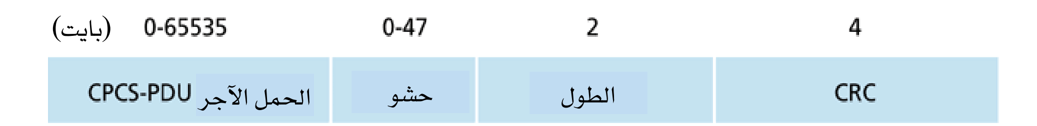
الخصائص الرئيسة لشبكات ATM

كما تقدم في الجزء 1-4، تدعم شبكات ATM عدّة نماذج خدمة، بما في ذلك خدمة معدّل البتات الثابت (Constant Bit Rate (CBR)، وخدمة معدل البتات المتغير (Variable Bit Rate (VBR)، وخدمة معدّل البتات المتوفر (Available Bit)

Rate (ABR)) ، وخدمة معدّل البتات غير المحدّد ((Unspecified Bit Rate (UBR)).
تعتمد ATM بنيةً معمارية للشبكة أساسها تحويل الرزم والدوائر الافتراضية ((Virtual Circuits (VCs)). تذكّر أننا سبق أن استعرضنا موضوع الدوائر الافتراضية بشيء من التفصيل في الجزء 4-2-1. تم تنظيم البنية المعمارية الكلية لشبكات ATM على شكل ثلاث طبقات كما هو مبين في الشكل 5-32.

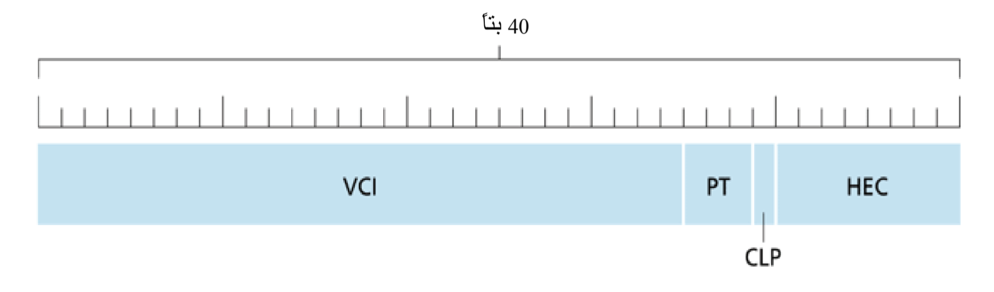
تشبه طبقة التكيّف بشبكة ((ATM Adaptation Layer AAL (ATM) تقريباً طبقة النقل في الإنترنت، وتوجد فقط في أجهزة ATM الموجودة على حافة الشبكة. على جانب الإرسال يتم تمرير البيانات إلى طبقة AAL من تطبيق أو بروتوكول في الطبقة الأعلى (مثل IP، إذا كانت شبكة ATM تستخدم لتشبيك أجهزة IP). على جانب الاستقبال ترفع طبقة AAL البيانات التي تم استلامها إلى البروتوكول أو التطبيق في الطبقة الأعلى. تم تعريف طبقات AAL مختلفة مثل AAL1 لخدمات معدّل البتات الثابت ومحاكاة الدوائر، وAAL2 لخدمات معدّل البتات المتغير (كالفيديو بمعدّل بتات متغير)، وAAL5 لخدمات البيانات (كنقل وحدات بيانات IP). من بين الخدمات التي تؤديها طبقة AAL اكتشاف الأخطاء، والتجزيء (segmentation) وإعادة التجميع (reassembly) . تعرف وحدة البيانات التي تتعامل معها طبقة AAL باسم عام هو وحدة بيانات بروتوكول AAL، وهي تكافئ تقريباً قطع بيانات UDP أو TCP.

يبين الشكل 5-33 وحدة بيانات بروتوكول AAL5. إن حقول وحدة البيانات بسيطة نسبياً. يضمن الحقل PAD أن وحدة البيانات تتكون من عدد صحيح من مضاعفات 48 بايتاً، لكي يسمح ذلك بتجزيء وحدة البيانات لتلائم حمولة بطول 48 بايتاً على رزم ATM التحتية (والتي تُعرف بخلايا ATM). يميّز حقل الطول حجم حمولة وحدة البيانات، بحيث يمكن إزالة الحقل PAD عند المُستقبل. يستخدم حقل CRC لاكتشاف أخطاء البتات بنفس أسلوب فحص الفائض الدوري المستخدم في الإيثرنت. يمكن أن يصل طول حقل الحمولة إلى 65535 بايت.



الشكل 5-33 وحدة بيانات بروتوكول AAL5.

دعنا الآن ننزل طبقة واحدة لأسفل لنتناول طبقة ATM، والتي تقع في قلب البنية المعمارية للشبكة. تعرّف طبقة ATM هيكل خلية ATM ومعنى كل حقل في الخلية. إن أهمية خلية ATM لشبكة ATM تماثل أهمية وحدة بيانات IP لشبكة IP. تشكّل البايتات الخمس الأولى من خلية ATM ترويسة ATM؛ بينما تشكّل البايتات الـ 48 الباقية حمولة ATM. يبين الشكل 5-34 صيغة ترويسة خلية ATM.



الشكل 5-34 صيغة ترويسة خلية ATM.

تؤدي الحقول المختلفة في خلية ATM الوظائف التالية:

- حقل معرّف القناة (أو الدائرة) الافتراضية (Virtual Channel Identifier (VCI)): يبين القناة الافتراضية التي تنتمي إليها الخلية. كما هو الحال في معظم تقنيات الشبكات التي تستخدم دوائر افتراضية، يتم ترجمة معرّف الخلية من وصلة إلى وصلة (انظر الجزء 4-2-1).
- حقل نوع الحمولة (Payload Type (PT)): يشير إلى نوع الحمولة الموجودة في خلية ATM . هناك عدة أنواع من حمولة البيانات، وعدة أنواع من حمولة

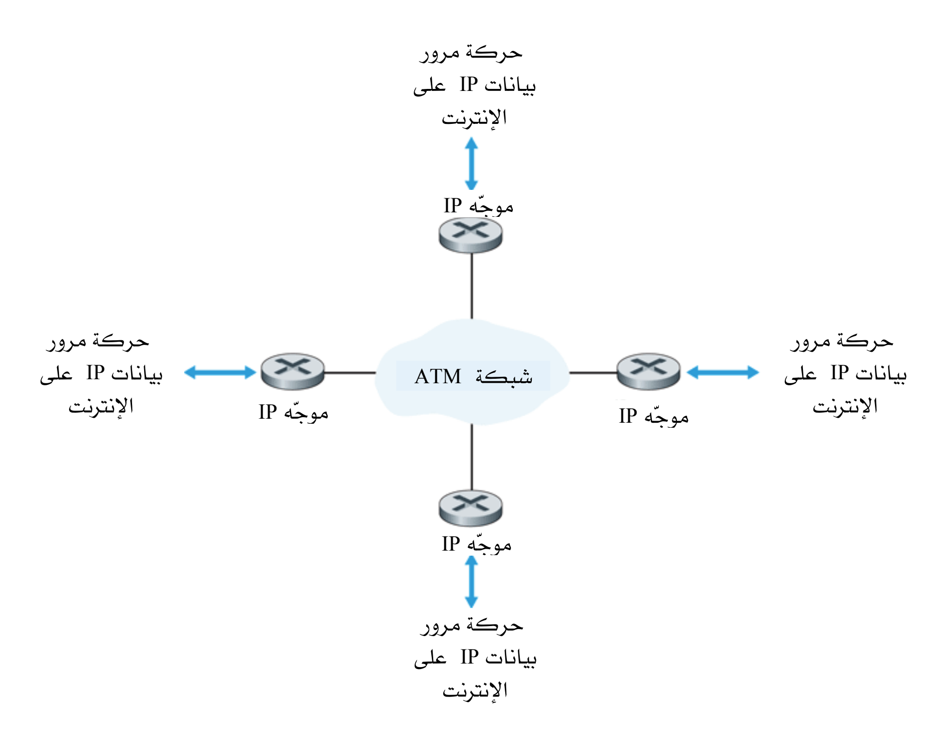
الصيانة، ونوع حمولة خلية شاغرة. يتضمّن حقل PT أيضاً بتاً لتمييز الخلية الأخيرة في وحدة بيانات بروتوكول AAL المجزّأة.

- بت أولوية الفقد للخلية (Cell-Loss Priority (CLP)). يمكن للمصدر إعطاؤها القيمة 1 للتفريق بين حركة مرور البيانات ذات الأولوية العالية وذات الأولوية المنخفضة. إذا حدث ازدحام وكان على محوّل ATM إهمال خلايا يمكن للمحوّل استخدام هذا البت للتخلص من حركة مرور البيانات ذات الأولوية المنخفضة.
- بايت التحكم في خطأ الترويسة (Header Error Control (HEC)). بتات اكتشاف الأخطاء التي تحمي ترويسة الخلية.

قبل أن يبدأ مصدر في إرسال خلايا إلى وجهة، يتعين على شبكة ATM أولاً تأسيس قناة افتراضية تمتد من المصدر إلى الوجهة. لا تعدو القناة الافتراضية كونها دائرة افتراضية كما وصفنا في الجزء 4-2-1. كل قناة افتراضية هي مسار يتألف من سلسلة وصلات بين المصدر والوجهة. يرتبط بكل وصلة على القناة الافتراضية مُعرّف القناة الافتراضية (VCI). في كل مرة يتم تأسيس أو فض قناة افتراضية، يتعين تحديث جداول الترجمة الخاصة بالقنوات الافتراضية (انظر الجزء 4-2-1). في حالة استخدام قناة افتراضية دائمة لن تكون هناك حاجة لتأسيس وفض القناة بطريقة ديناميكية. عند الحاجة لتأسيس وفض القناة بطريقة ديناميكية يوفر البروتوكول Q.2931 [Black 1997; ITU-T Q.2931 1994] عمليات التأشير اللازمة بين المحوّلات والأنظمة الطرفية في شبكة ATM.

تقع طبقة ATM المادية في أسفل القاع من رصة بروتوكولات ATM، وتتعامل مع الفولطيات، وتوقيت البتات، وتأطير البيانات على الوسط المادي. يعتمد جزء كبير من الطبقة المادية على خصائص الوصلة المادية. هناك صنفان أساسيان من الطبقات المادية: الطبقات التي لها هيكل محدد لإطار الإرسال (مثل: T1 وT3 وSONET وSDH)، والطبقات التي ليس لها هيكل محدد لإطار الإرسال. إذا كان للطبقة المادية هيكل إطار، تكون الطبقة مسؤولة عن توليد وتحديد الإطارات. ينبغي عدم الخلط بين كلمة "إطارات" المستخدمة هنا واستخدامنا لها في سياق

طبقة ربط البيانات (كما في الإيثرنت). يمثل إطار الإرسال هنا آلية خاصة بالطبقة المادية لتنظيم البتات المُرسِلة، كما في حالة إطارات الإرسال المتعدد بتقسيم الزمن (TDM).



الشكل 5-35 شبكة ATM في قلب العمود الفقري للإنترنت.

تشغيل بروتوكول IP فوق شبكة ATM

دعنا الآن ندرس كيف يمكن استخدم شبكة ATM للتشبيك ما بين أجهزة IP. يبين الشكل 5-35 شبكة عمود فقري ATM بأربع نقاط دخول وخروج لحركة بيانات الإنترنت IP. لاحظ أن كل نقطة دخول وخروج هي موجّه. يمكن أن تمتد شبكة العمود الفقري ATM لتغطي قارة بأكملها وتتضمن العشرات بل المئات من محوّلات ATM. تستخدم معظم شبكات الأعمدة الفقرية من نوع ATM قناة افتراضية دائمة بين كل زوج من نقاط الدخول والخروج. باستخدام قنوات افتراضية

دائمة يمكن توجيه خلايا ATM من نقطة دخول إلى نقطة خروج دون الحاجة لتأسيس أو فض قنوات افتراضية بطريقة ديناميكية. غير أن استخدام قنوات افتراضية دائمة يكون ممكناً فقط عندما يكون عدد نقاط الدخول والخروج قليلاً نسبياً. للتوصيل مباشرة بين n نقطة دخول وخروج نحتاج لـ $n(n - 1)$ قناة افتراضية دائمة.

ستحتاج كل واجهة على موجّه موصّل بشبكة ATM إلى عنوانين، تقريباً بنفس الطريقة التي يحتاج بها مضيف IP إلى عنوانين لوصلة إيثرنت: عنوان IP وعنوان ماك. بالمثل، يكون لواجهة ATM عنوان IP وعنوان ATM. خذ في الاعتبار الآن وحدة بيانات IP تعبر شبكة ATM المبينة في الشكل 5-35. في الحالة الأبسط تبدو شبكة ATM كوصلة منطقية واحدة تربط تلك الموجّهات الأربعة كما في حالة استخدام الإيثرنت لتوصيل أربعة موجّهات. دعنا نشير إلى الموجّه الذي تدخل منه أي وحدة بيانات إلى شبكة ATM بـ "موجّه دخول" والموجّه الذي تغادر منه وحدة البيانات الشبكة بـ "موجّه خروج". يقوم موجّه الدخول بما يلي:

1. فحص عنوان الوجهة لوحدة البيانات.
2. الدخول على جدول التوجيه لديه وتحديد عنوان IP لموجّه الخروج (أي الموجّه التالي في طريق وحدة البيانات).
3. لتوصيل وحدة البيانات إلى موجّه الخروج، يتعامل موجّه الدخول مع ATM كمجرد بروتوكول طبقة ربط بيانات آخر. لنقل وحدة البيانات إلى الموجّه التالي، علينا تحديد العنوان المادي لموجّه القفزة التالية. تذكّر من مناقشتنا في الجزء 5-4-2 إن هذا يتم باستخدام بروتوكول تحويل العناوين ARP. في حالة واجهة ATM يفحص موجّه الدخول جدول ATM ARP مستخدماً عنوان IP لموجّه الخروج ليحصل على عنوان ATM لموجّه الخروج. يوجد وصف لبروتوكول ATM ARP في [RFC 2225].
4. يقوم بروتوكول IP في موجّه الدخول بعد ذلك بتمرير وحدة البيانات مع عنوان ATM لموجّه الخروج إلى طبقة ربط البيانات بشبكة ATM.

بعد الانتهاء من تلك الخطوات الأربع، تخرج مهمة نقل وحدة البيانات إلى موجّه الخروج من أيدي بروتوكول IP وتنتقل إلى أيدي بروتوكول ATM. على ATM الآن نقل وحدة البيانات إلى عنوان ATM للوجهة والذي تم الحصول عليه في الخطوة 3 أعلاه. تنضوي تحت هذه المهمّة مهمّتان ثانويتان:

1. تحديد المعرّف VCI للقناة الافتراضية التي تؤدي إلى عنوان ATM للوجهة.
2. تجزئة وحدة البيانات إلى خلايا في جانب الإرسال على القناة الافتراضية (أي على موجّه الدخول)، ثم إعادة تجميع الخلايا للحصول على وحدة البيانات الأصلية في جانب الاستقبال على القناة الافتراضية (أي على موجّه الخروج).

المهمّة الثانوية الأولى سهلة. تحتوي الواجهة في جانب الإرسال على جدول للتحويل من عناوين ATM إلى معرّفات القنوات الافتراضية المناظرة. ونظراً لأننا افترضنا إن القنوات الافتراضية دائمة، فسيكون ذلك الجدول ثابتاً ومحدّثاً (بينما إذا كانت القنوات الافتراضية غير دائمة، فسيُستخدم بروتوكول التأشير ATM Q.2931 لتأسيس وفض القنوات الافتراضية بشكلٍ ديناميكي). أما المهمّة الثانية فتستحق تناولاً أكثر حذراً. أحد الطرق هو استخدام تجزئة IP كما تناولناه في الجزء 4-4. في هذه الحالة يقوم موجّه الإرسال أولاً بتجزئة وحدة البيانات الأصلية إلى أجزاء، بحيث لا يزيد كل جزء عن 48 بايتاً، ليتسنى وضع كل جزء كحمولة في خلية ATM. لكن هذه الطريقة في التجزئة تعاني من مشكلة كبيرة ـ فكل جزء IP له عادةً ترويسة تتكون من 20 بايتاً، وعليه فإن كل خلية ATM تحمل جزء IP ستحمل فقط 28 بايتاً من المعلومات المفيدة مقابل 25 بايتاً من العبء الإضافي (overhead). لهذا السبب تستخدم ATM بروتوكول AAL5 لتجزئة وإعادة تجميع وحدات البيانات بطريقة أكثر كفاءة.

بعد ذلك تنقل طبقة ATM كل خلية عبر الشبكة إلى عنوان ATM للوجهة. عند كل محوّل ATM بين مصدر ATM ووجهة ATM، يتم معالجة الخلية بواسطة طبقة ATM المادية وطبقات ATM الأخرى باستثناء طبقة AAL. في كل محوّل، يتم عادةً ترجمة معرّف القناة الافتراضية VCI (راجع الجزء 4-2-1) ويعاد حساب بايت التحكم في خطأ الترويسة (HEC). عندما تصل الخلايا إلى عنوان ATM للوجهة يتم

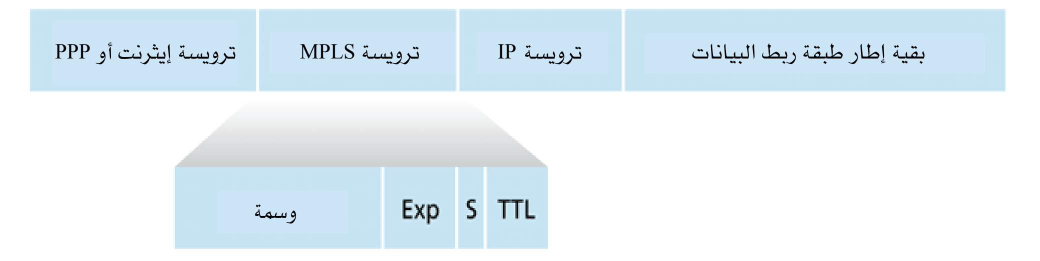
توجيهها إلى مخزن AAL مؤقت تم تخصيصه للقناة الافتراضية المستخدمة. بعد ذلك يعاد بناء وحدة بيانات بروتوكول AAL5 واستخراج وحدة بيانات IP وتمريرها عبر رصة البروتوكولات إلى طبقة IP.

5-8-2 تقنية تحويل الوسمة متعدد البروتوكول (MPLS)

تم تطوير تقنية تحويل الوسمة متعدد البروتوكول (Multi-Protocol Label Switching (MPLS)) من خلال جهود الصناعة في أواسط التسعينيات إلى أواخرها من أجل تحسين سرعة التوجيه في موجّهات IP، وذلك بتبنّي مفهوم أساسي من عالم شبكات الدائرة الافتراضية: استخدام وسمة (label) بطول ثابت. لم يكن الهدف الاستغناء عن البنية التحتية لتوجيه وحدات بيانات IP والمبني على أساس معرفة عنوان الوجهة النهائية واستبداله ببنية أخرى أساسها وسمات بأطوال ثابتة ودوائر افتراضية، ولكنه كان إدخال تحسينات على الوضع الحالي لنظام توجيه IP بوسم وحدات بيانات IP بشكلٍ اختياري والسماح للموجّهات بتوجيه حزم البيانات بناءً على الوسمات ثابتة الطول (بدلاً من عناوين IP للوجهة النهائية) كلما كان ذلك ممكناً. من المهم ملاحظة أن هذه الأساليب تعمل بالتعاون يداً بيد مع بروتوكول IP، مستخدمةً أنظمة IP للعنونة والتوجيه. قام فريق عمل هندسة الإنترنت بتوحيد تلك الجهود في بروتوكول MPLS [RFC 3031; RFC 3032]، والذي يدمج أساليب الدوائر الافتراضية في سياق شبكات توجيه وحدات البيانات.

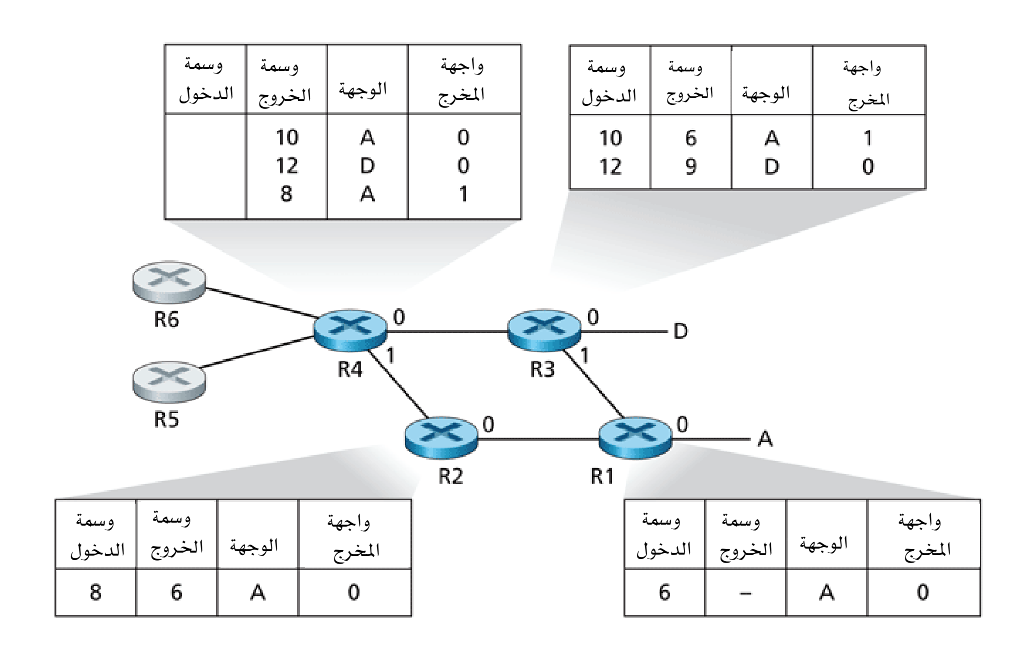
دعنا نبدأ دراستنا لبروتوكول MPLS باستعراض صيغة إطار طبقة ربط البيانات التي يعالجها موجّه مزوّد بإمكانيات التعامل مع MPLS. كما هو موضّح في الشكل 5-36، يتضمن إطار طبقة ربط البيانات المُرسَل على وصلة PPP أو شبكة محلية (كالإيثرنت) ترويسة MPLS صغيرة تضاف بين ترويسة الطبقة 2 (أي PPP أو الإيثرنت) وترويسة الطبقة 3 (أي IP). يُعرّف طلب الاقتراحات RFC 3032 صيغة ترويسة MPLS لتلك الوصلات؛ كما تم تعريف ترويسات MPLS لشبكات ATM وشبكات ترحيل الإطارات في وثائق RFC أخرى. تضم حقول ترويسة MPLS حقل الوسمة (label) (والتي تلعب دور مُعرّف الدائرة الافتراضية (VCI) الذي

استخدمناه في الجزء 4-2-1)، وحقلاً بطول 3 بتات محجوزة للاستعمال التجريبي، وحقلاً من بت واحد S يُستخدم للإشارة إلى نهاية سلسلة من ترويسات MPLS المرصوصة (stacked) (وهو موضوع متقدّم لن نغطيه هنا)، وأخيراً حقل يبين فترة العمر (TTL).



الشكل 5-36 ترويسة MPLS: والتي تقع بين ترويسة طبقة ربط البيانات وترويسة طبقة الشبكة.

يتضح مباشرةً من الشكل 5-36 أن الإطار المُحسّن بـ MPLS يمكن تبادله فقط بين موجّهين يكون لكلٍّ منهما القدرة على التعامل مع MPLS (حيث إن الموجّهات التي لا تفهم MPLS سترتبك تماماً عندما تجد ترويسة MPLS حيث تتوقّع وجود ترويسة IP!). غالباً ما يُطلق على الموجّه المزوّد بإمكانيات MPLS اسم موجّه التحويل بوسمة (label-switched router)، حيث إنه يقوم بتوجيه إطار MPLS باستخدام قيمة حقل الوسمة في ترويسة MPLS للدخول على جدول التوجيه الموجود عليه، ثم تمرير وحدة البيانات فوراً إلى واجهة الخرج المناسبة. وهكذا فإن الموجّه المزوّد بإمكانيات MPLS لا يحتاج لاستخراج عنوان IP للوجهة ثم تحديد مطابقة أطول بادئة بالرجوع إلى جدول التوجيه لديه تمهيداً لتوجيه الإطار. لكن كيف يعرف موجّه في الواقع ما إذا كان الموجّه المجاور له مجهزاً فعلاً للتعامل مع MPLS، وكيف يعرف الموجّه أي وسمة تقابل عنوان IP مُعطى لوجهة نهائية؟ للإجابة على هذه الأسئلة، نحتاج لإلقاء نظرة على التفاعل ما بين مجموعة موجّهات مجهزة للتعامل مع MPLS.



الشكل 5-37 توجيه محسّن لوحدات بيانات IP باستخدام وسمات MPLS.

في المثال الموضح في الشكل 5-37، للموجّهات من R1 إلى R4 القدرة على التعامل مع MPLS، بينما R5 وR6 موجّهان IP عاديّان. افترض أن الموجّه R1 أعلن للموجّه R2 بأنه (أي R1) يمكنه أن يوجّه إلى الوجهة A، وأن الإطارات التي يتم استلامها بوسمات MPLS قيمتها 6 سترسل إلى الوجهة A. وكذلك أعلن الموجّه R3 لـ R4 بأنه يمكنه أن يوجّه إلى الوجهتين A وD، وأن الإطارات القادمة بوسمات MPLS قيمتها 10 و12 ستوجّه إلى هاتين الوجهتين على الترتيب. كما أعلن الموجّه R2 أيضاً للموجّه R4 بأنه يمكنه أن يصل إلى الوجهة A، وأن الإطارات بوسمات MPLS قيمتها 8 سيتم تحويلها نحو A. لاحظ أن الموجّه R4 أصبح الآن في وضعٍ فريد، حيث يتوافر لديه مساران من مسارات MPLS للوصول إلى A ـ عن طريق الواجهة 0 بوسمة خروج قيمتها 10، وعن طريق الواجهة 1 بوسمة خروج قيمتها 8. الصورة العامة التي نخرج بها من الشكل 5-37 هي أن أجهزة IP (الموجّهين R5 وR6 والمضيفين A وD) ترتبط مع بعضها عن طريق بنية تحتية بتقنية MPLS (الموجّهات من R1 إلى R4 والمزودة بإمكانيات MPLS)، تقريباً بنفس الطريقة التي يمكن بها

لشبكة بيانات محلية أو شبكة ATM تشبيك أجهزة IP سوية. وكما في حالة شبكة محلية محوّلة أو شبكة ATM، فإن الموجّهات من R1 إلى R4 والمزودة بإمكانيات MPLS تؤدي ذلك بدون أن تتعامل أبداً مع ترويسة IP في قطعة البيانات.

في مناقشتنا أعلاه، لم نحدّد البروتوكول المعيّن المستخدم في توزيع وسمات MPLS بين الموجّهات المزودة بإمكانيات MPLS؛ نظراً لأن تلك التفاصيل تقع خارج نطاق هذا الكتاب. ولكن نلاحظ هنا أن مجموعة العمل المنبثقة من IETF والمختصة بـ MPLS قد حدّدت في [RFC 3468] أن امتداداً لبروتوكول RSVP (والذي سندرسه في الفصل السابع)، ويعرف بـ RFC [RSVP-TE 3209] سيشكّل بؤرة الجهود لنظام التأشير باستخدام MPLS. وعليه فإننا نشجع القارئ المهتم بمراجعة RFC 3209.

حتى الآن ركزت مناقشتنا على أن MPLS يقوم بعملية التحويل بناءً على الوسمات، بدون حاجة لأخذ عنوان IP لوحدة البيانات في الاعتبار. غير أن الفوائد الحقيقية لـ MPLS والسبب وراء الاهتمام الكبير به حالياً لا يكمن في الزيادة المحتملة في سرعة تحويل الرزم فقط، ولكن بالأحرى في الإمكانيات الجديدة لإدارة حركة مرور البيانات والتي يوفرها MPLS. كما لاحظنا أعلاه، يتوافر للموجّه R4 مساران MPLS إلى المضيف A، إذا تم توجيه الرزم في طبقة IP الأعلى بناءً على عنوان IP، ستحدد بروتوكولات IP للتوجيه – والتي درسناها في الفصل الرابع – مساراً واحداً إلى A هو المسار الأقل كلفة. وهكذا يوفر MPLS إمكانية توجيه الرزم عبر مسارات قد لا تكون متاحة عند استخدام بروتوكولات توجيه IP القياسية. يعتبر هذا مجرد شكل واحد بسيط فقط من تطبيقات هندسة مرور البيانات الممكنة باستخدام MPLS [RFC 3346; RFC 3272; RFC 2702; Xiao 2000]، حيث يمكن لمشغّل الشبكة أن يتخطى توجيه IP المعتاد ويُجبر بعض حركة المرور المرسَلة إلى وجهةٍ ما على سلوك مسار بعينه، وحركة مرور أخرى إلى نفس الوجهة على سلوك مسار آخر (سواء لأسباب تتعلق بسياسة المرور، أو الأداء، أو أي سبب آخر).

يمكن أيضاً استخدام MPLS للعديد من الأغراض الأخرى، كالاستعادة السريعة لمسارات توجيه MPLS. مثلاً لإعادة توجيه المرور عند حدوث عطل في وصلة إلى مسار احتياطي محسوب مسبقاً [Kar 2000; Huang 2002; RFC 3469]. يمكن أيضاً استخدام MPLS لتحقيق هيكل الخدمة التفاضلية ("DiffSev") والتي سندرسها في الفصل السابع. وأخيراً نلاحظ أن MPLS يمكن أن يُستخدم أيضاً لتطبيق ما يسمّى بالشبكة الافتراضية الخاصة ((Virtual Private Network (VPN)) حيث يستخدم موفر خدمة الإنترنت شبكته المزوّدة بإمكانيات MPLS في توصيل الشبكات المختلفة الخاصة بعميلٍ ما ببعضها البعض، وبذلك يمكن عزل كلٍّ من الموارد، والعنونة المستخدمة بواسطة شبكة VPN للعميل عن المستخدمين الآخرين الذين يعبرون شبكة موفر الخدمة. لمزيد من التفاصيل انظر [DeClercq 2002].

لقد كانت مناقشتنا لـ MPLS مختصرة بالضرورة، ولذا فنحن نشجّعك على الرجوع إلى المراجع التي ذكرناها للحصول على المزيد من التفاصيل. نلاحظ أنه مع ظهور العديد من الاستخدامات الممكنة لـ MPLS، يبدو أن هذا الأسلوب الجديد سيوفر حلاً للكثير من المشاكل في مجال هندسة حركة مرور الإنترنت!

5-9 الخلاصة

تناولنا في هذا الفصل طبقة ربط البيانات، حيث استعرضنا خدماتها، والمبادئ التي تحكم عملها، وعدداً من البروتوكولات المحددة والهامة التي تستخدم تلك المبادئ في تحقيق خدمات طبقة ربط البيانات.

رأينا أن الخدمة الأساسية لطبقة ربط البيانات تتلخص في نقل وحدة بيانات طبقة الشبكة من عقدة (موجّه أو مضيف) إلى عقدة مجاورة. وعرفنا أن كل بروتوكولات طبقة ربط البيانات تقوم بتغليف وحدة بيانات طبقة الشبكة ضمن إطار طبقة ربط البيانات قبل إرسال الإطار على الوصلة إلى العقدة المجاورة. وباستثناء وظيفة التأطير المشتركة تلك، وجدنا أن بروتوكولات طبقة ربط البيانات المختلفة توفر خدمات وتستخدم طرقاً مختلفة جداً للوصول للوصلة، ولتوصيل البيانات (الموثوقية واكتشاف وتصحيح الأخطاء)، ولضبط التدفق، وللإرسال

(مثلاً إرسال مزدوج تماماً أو نصف مزدوج). من أسباب هذه الاختلافات كثرة الأنواع المختلفة من الوصلات التي يتعيّن أن تعمل عليها بروتوكولات طبقة ربط البيانات. فوصلة نقطة إلى نقطة مثلاً وصلة بسيطة لها مُرسِل واحد ومُستقبِل واحد يتصلان عبر "سلك" واحد. أما وصلة الوصول المتعدد فمشتركة بين العديد من المُرسِلين والمُستقبِلين. لذلك فإن طبقة ربط البيانات لقناة وصول متعدد لها بروتوكول (هو بروتوكول الوصول المتعدد) لتنسيق الوصول للوصلة بين عدة مستخدمين. في حالة شبكات ATM وMPLS يمكن في الواقع أن تكون الوصلة التي تصل بين عقدتين متجاورتين (على سبيل المثال موجّهي IP متجاورين من منظور IP، أي يفصل بينهما قفزة واحدة على المسار نحو وجهةٍ ما) شبكة في حد ذاتها. من وجهة نظر معينة ينبغي ألاّ تبدو فكرة اعتبار الشبكة كوصلة فكرة مستغربة. فعلى سبيل المثال خطّ الهاتف الذي يوصل مودم بحاسب بيتي إلى مودم بموجّه بعيد هو في الحقيقة مسار عبر شبكة هاتف متطوّرة ومعقّدة.

تناولنا بعض المبادئ التي ينبني عليها الاتصال عبر طبقة ربط البيانات، ومنها: أساليب اكتشاف وتصحيح أخطاء البيانات، وبروتوكولات الوصول المتعدد، وعنونة طبقة ربط البيانات، وبناء شبكات بيانات محلية ممتدة باستخدام المجمِّعات والمحوّلات. أما فيما يتعلق باكتشاف وتصحيح الأخطاء، فقد رأينا كيف أن إلحاق بتات إضافية بترويسة إطار البيانات تمكننا من اكتشاف - وفي بعض الحالات تصحيح - أخطاء البتات التي قد تطرأ على الإطار أثناء انتقاله على الوصلة. كما غطّينا الأساليب البسيطة التي تستخدم بتات التكافؤ والمجموع التدقيقي، بالإضافة إلى أسلوب فحص الفائض الدوري الأكثر متانة. وانتقلنا بعد ذلك إلى موضوع بروتوكولات الوصول المتعدد، حيث درسنا ثلاثة طرق رئيسة لتنسيق الوصول إلى قناة إذاعة مشتركة: تقسيم القناة (مثل TDM و FDM)، والوصول العشوائي (مثل بروتوكولات ALOHA وبروتوكولات CSMA)، وأساليب التناوب على القناة (كأساليب الاستفتاء وتمرير العلامة). رأينا أنه نتيجة لجعل عدة عقد تشترك في قناة إذاعة واحدة، ظهرت الحاجة لعناوين العقد في طبقة ربط البيانات. كما عرفنا كيف أن العناوين المادية تختلف كثيراً عن عناوين طبقة الشبكة،

وأنه في حالة الإنترنت يُستخدم بروتوكول خاص (بروتوكول تحويل العناوين ARP) للترجمة بين هذين النوعين من العناوين. تناولنا بعد ذلك كيف تشكّل العقد التي تشترك في قناة إذاعة شبكة محلية، وكيف يمكن توصيل عدد من تلك الشبكات المحلية لتكوين شبكات محلية أكبر، كل ذلك بدون اللجوء إلى استخدام بروتوكولات التوجيه في طبقة الشبكة لتشبيك تلك العقد المحلية.

غطّينا أيضاً عدداً من البروتوكولات المحددة لطبقة ربط البيانات بالتفصيل، كبروتوكول الإيثرنت وبروتوكول PPP. ثم أنهينا دراستنا لطبقة ربط البيانات بالتركيز على كيفية قيام شبكات ATM وMPLS بخدمات طبقة ربط البيانات عند تشبيك موجّهات IP.

وبعد أن انتهينا من تغطية طبقة ربط البيانات تكون رحلتنا عبر رصة البروتوكولات قد انتهت! بالتأكيد تحت طبقة ربط البيانات توجد الطبقة المادية، ولكننا نرى أنه من الأفضل ترك تفاصيل الطبقة المادية لمقرر دراسي آخر (مثلاً في نظرية الاتصالات بدلاً من شبكات الحاسب). علماً بأننا مع ذلك قد لمسنا عدداً من جوانب الطبقة المادية في هذا الفصل (كمناقشتنا القصيرة لتشفير مانشستر في الجزء 5-5) وفي الفصل الأول (كمناقشتنا لأوساط النقل المادية في الجزء 1-2). سنأخذ الطبقة المادية بعين الاعتبار مرةً أخرى عند دراستنا لخصائص وصلة اللاسلكي في الفصل القادم.

وبالرغم من أن رحلتنا عبر رصة البروتوكولات قد انتهت، إلا أن دراستنا لشبكات الحاسب لم تنتهِ بعد. في الفصول الأربعة التالية سنغطّي الشبكات اللاسلكية، وشبكات الوسائط المتعددة، وأمن الشبكات، وإدارة الشبكات. لا ينضوي أيٌّ من هذه المواضيع الأربعة تحت طبقة واحدة. ففي الواقع يتوزع كل موضوع منها على عدة طبقات. لذلك فإن فهم تلك المواضيع (والتي وُصفت بكونها "مواضيع متقدّمة" في بعض كتب الشبكات) يتطلّب أساساً متيناً في كل طبقات رصة البروتوكولات ـ وهي المهمة التي قد انتهينا منها الآن عندما أكملنا دراستنا لطبقة ربط البيانات!

أسئلة وتمارين وتدريبات الفصل الخامس

❖ أسئلة مراجعة

• الأجزاء 5-1 و 5-2

1. لو أن كل الوصلات في الإنترنت وفّرت خدمة نقل موثوقة للبيانات، هل يعني ذلك أنه لن يكون هناك داعٍ لاستخدام خدمة TCP للنقل الموثوق؟ علل إجابتك.
2. اذكر بعض الخدمات التي يمكن لبروتوكول طبقة ربط البيانات توفيرها لطبقة الشبكة؟ أيٌّ من تلك الخدمات له نظير في بروتوكول IP؟ وبروتوكول TCP؟

• الجزء 5-3

3. افترض أن عقدتين تبدآن في إرسال رزمة طولها L بتاً في نفس الوقت على قناة إذاعة بمعدل R بت/ثانية. ليكن d_{prop} هو تأخير الانتقال بين العقدتين. هل سيحدث اصطدام لو كان $d_{prop} < L/R$؟ علل إجابتك.
4. في الجزء 5-3، ذكرنا أربع خواص مطلوبة في قناة الإذاعة. أي تلك الخواص تتوفر في بروتوكول ألوها الشرائحي؟ أي تلك الخواص تتوفر في بروتوكول تمرير العلامة؟
5. صف بروتوكولي الاستطلاع وتمرير العلامة مع التشبيه بتفاعل الناس في حفل.
6. لماذا يعاني بروتوكول تمرير العلامة من انخفاض في كفاءته إذا كانت الشبكة المحلية تغطي منطقة جغرافية كبيرة؟

• الجزء 5-4

7. ماحجم حيز عنوان الماك؟، وعنوان IPv4؟، وعنوان IPv6؟
8. افترض أن كلاً من العقد A وB وC موصّلة بنفس شبكة إذاعة محلية (LAN) عن طريق موائم الشبكة الخاص بها. إذا أرسلت A الآلاف من قطع بيانات IP إلى B يتضمن كل إطار من الإطارات التي تغلف تلك القطع عنوان الماك الخاص بالعقدة . هل سيمرر موائم الشبكة الخاص بالعقدة C قطع بيانات IP ضمن تلك الإطارات إلى العقدة C؟ كيف ستتغير إجابتك إذا كانت A ترسل تلك الإطارات على عنوان الماك المخصص للإذاعة؟

9. لماذا يُرسل استفسار ARP ضمن إطار إذاعة؟ لماذا تُرسل إجابة ARP ضمن إطار يحمل عنوان الماك لوجهة محددة؟
10. للشبكة المبينة في الشكل 5-19، يتضمن الموجّه وحدتي ARP، لكلٍّ منهما جدول ARP الخاص بها. هل يمكن أن يظهر عنوان الماك نفسه في كلا الجدولين؟

- **الجزء 5-5**

11. قارن بين صيغة إطار الإيثرنت في كلٍّ من 10BASE-T و 100BASE-T والإيثرنت بسرعة جيجابت/ثانية.
12. في بروتوكول CSMA/CD، بعد خامس اصطدام، ماهو احتمال أن تختار عقدة K = 4؟ ماهو التأخير بالثانية المناظر لتلك القيمة لـ K على إيثرنت سرعتها 10 ميجابت/ثانية؟

- **الجزء 6-5**

13. بالرجوع إلى الشكل 5-26، كم عدد الشبكات الفرعية الموجودة، آخذاً في الاعتبار طريقة العنونة الواردة في الجزء 4-4؟

❖ تدريبات

1. افترض أن محتوى المعلومات في رزمة ما هو 1010101010101011 وأن نظام تكافؤ زوجي يجري استخدامه. ماهي قيمة الحقل الذي يتضمن بتات التكافؤ في حالة اتباع نظام في بعدين؟ يجب أن تكون إجابتك بحيث يُستخدم حقل المجموع التدقيقي بأقل طول ممكن.
2. بيّن (مستخدماً مثالاً غير المثال الموضح في الشكل 5-6) أن فحص التكافؤ ببعدين يمكن أن يكتشف ويصحح خطأً في بت واحد. بيّن مع التمثيل أن خطأ في بتّين سيمكن اكتشافه ولكن لن يمكن تصحيحه.
3. افترض أن جزء المعلومات في رزمة (D في الشكل 5-4) يضم 10 بايتات تمثل القيمة الثنائية للأعداد الصحيحة من 0 إلى 9. احسب المجموع التدقيقي للإنترنت لتلك البيانات.
4. خذ في الاعتبار التمرين السابق، ولكن بدلاً من احتواء البيانات على القيمة الثنائية للأعداد الصحيحة من 0 إلى 9، افترض أن البايتات العشرة تتضمن:

a. القيم الثنائية للأعداد من 1 إلى 10.
b. تمثيل الحروف الكبيرة من A إلى J بصيغة ASCII.
c. تمثيل الحروف الصغيرة من a إلى j بصيغة ASCII.

احسب المجموع التدقيقي للإنترنت لتلك البيانات.

5. خذ في الاعتبار المولد G من أربعة بتات والمبين في الشكل 5-8. بافتراض أن قيمة D هي 10101010، ما هي قيمة R؟

6. خذ في الاعتبار التمرين السابق، ولكن مع افتراض أن D لها القيمة:
   a. 10010001.
   b. 10100011.
   c. 01010101.

7. في الجزء 5-3 استعرضنا طريقة اشتقاق تعبير رياضي لكفاءة بروتوكول ألوها الشرائحي. في هذا التمرين سنكمل الاشتقاق.
   a. تذكّر أنه في وجود N عقدة نشطة، تكون كفاءة بروتوكول ألوها الشرائحي هي $Np(1 - p)^{N-1}$. أوجد قيمة p التي تجعل قيمة هذا التعبير الرياضي نهاية عظمى.
   b. باستخدام قيمة p التي حصلَتَ عليها في الجزء (a) أعلاه من هذا السؤال، أوجد كفاءة بروتوكول ألوها الشرائحي بجعل N تقارب مالانهاية. ملاحظة: $(1-1/N)^N$ تقارب $(1/e)$ عندما تقارب N ما لانهاية.

8. بيّن أن الكفاءة القصوى لبروتوكول ألوها الأصلي هي $(1/2e)$. ملاحظة: هذا التمرين سهل إذا كنت قد أكملت التمرين السابق!

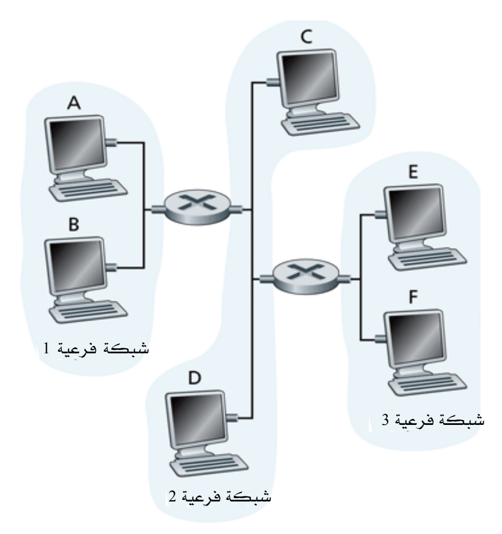
9. افترض أن العقد الثلاث A وB وC تتنافس فيما بينها للوصول إلى قناة باستخدام بروتوكول ألوها الشرائحي. افترض أن كل عقدة لديها عدد لا نهائي من الرزم تود إرسالها. تحاول كل عقدة الإرسال في كل شريحة زمنية بالاحتمال p. يطلق على الشريحة الأولى شريحة 1، والثانية شريحة 2، وهكذا.
   a. ما هو احتمال نجاح العقدة A في الإرسال في أول محاولة في الشريحة 4؟
   b. ما هو احتمال نجاح العقدة أي عقدة (A أو B أو C) في الإرسال في الشريحة 2؟
   c. ما هو احتمال حدوث أول نجاح في الشريحة 4؟
   d. ماهي كفاءة هذا النظام الذي يضم ثلاث عقد؟

10. مثّل بالرسم كفاءة بروتوكول ألوها الشرائحي وبروتوكول ألوها الأصلي كدالة في p للقيم التالية لعدد العقد النشطة N في الحالات التالية:
   a. $N = 10$
   b. $N = 25$

c. $N = 50$

11. خذ في الاعتبار قناة إذاعة عليها N عقدة ولها معدل إرسال R بت/ثانية. افترض أن قناة الإذاعة تستخدم أسلوب الاستطلاع (بإضافة عقدة استطلاع خاصة) لتنظيم الوصول المتعدد. افترض أن الفترة الزمنية من انتهاء عقدة من الإرسال إلى السماح للعقدة التالية بالإرسال (أي تأخير الاستطلاع) هي d_{poll}. افترض أنه أثناء دورة استطلاع يتم السماح لعقدة بإرسال Q بتاً كحد أقصى. ما هي طاقة الإرسال الإنتاجية القصوى لقناة الإذاعة تلك؟

12. خذ في الاعتبار الشبكات المحلية الثلاث الموصّلة فيما بينها عن طريق موجّهين، كما هو مبين في الشكل 5-38.



الشكل 5-38 ثلاث شبكات فرعية موصّلة فيما بينها عن طريق موجّهين.

a. أعد رسم الشكل بحيث يتضمن موائمات الشبكة

b. عيّن عناوين IP لكل الواجهات. استخدم عناوين بالصيغة 111.111.111.xxx للشبكة الفرعية 1، و عناوين بالصيغة 122.222.222.xxx للشبكة الفرعية 2، و عناوين بالصيغة 133.333.333.xxx للشبكة الفرعية 3.

c. عيّن عناوين الماك لكل الواجهات.

d. خذ في الاعتبار إرسال قطعة بيانات IP من المضيف A إلى المضيف F. افترض أن كل جداول ARP محدّثة تماماً. اذكر جميع الخطوات اللازمة على نسق ما قمنا به في حالة موجّه واحد في الجزء 5-4-2.

e. كرر الجزء د أعلاه مع افتراض أن جدول ARP للمضيف المرسِل فارغ بينما جداول ARP الأخرى محدّثة تماماً.

13. خذ في الاعتبار التمرين السابق، ولكن افترض أن الموجّه بين الشبكة الفرعية 2 والشبكة الفرعية 3 قد تم استبداله بمحوّل. قم بالإجابة على الأسئلة a إلى e في التمرين السابق في هذا السياق الجديد.

14. تذكّر أنه في بروتوكول الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام CSMA/CD ينتظر موائم الشبكة لمدة $K \times 512$ فترة بت بعد كل اصطدام، حيث K رقم يتم سحبه عشوائياً. في حالة $K = 100$، ماهي المدة التي ينتظرها الموائم قبل العودة للخطوة 2 (انظر البروتوكول) وذلك في حالة إيثرنت سرعتها 10 ميجابت/ثانية؟، وكذلك في حالة إيثرنت سرعتها 100 ميجابت/ثانية؟

15. افترض أن العقدتين A وB تقعان على نفس ناقل الإيثرنت بسرعة 10 ميجابت/ثانية، وأن تأخير الانتقال بين العقدتين هو 225 فترة بت. افترض أن العقدة A تبدأ بإرسال إطار، وقبل أن تنتهي بدأت العقدة B في إرسال إطار. هل يمكن للعقدة أ الانتهاء من إرسال إطارها قبل اكتشاف أن ب أخذت في الإرسال؟ علل إجابتك. إذا كانت الإجابة بنعم، فستظن أ مخطئةً أنها قد أرسلت إطارها بدون مشاكل. ملاحظة: افترض أنه عند الوقت t = صفر فترة بت، تبدأ A في إرسال إطار. في أسوأ الاحتمالات ترسل A إطاراً له أقل طول وقدره 512+64 فترة بت. وعليه ينبغي أن تنهي A إرسالها في اللحظة 512+64 فترة بت. ومن ثم فالإجابة تكون لا إذا وصلت إشارة B إلى A قبل الوقت 512+64 فترة بت. في أسوأ الاحتمالات، متى تصل إشارة B إلى A؟

16. افترض أن العقدتين A وB تقعان على نفس ناقل الإيثرنت بسرعة 10 ميجابت/ثانية، وأن تأخير الانتقال بين العقدتين هو 225 فترة بت. افترض أن كلاً من العقدتين A وB تبدآن بإرسال إطار في نفس الوقت، ويصطدم الإطاران، وتختار كلٌّ من العقدتين قيمة مختلفة لـ K في خوارزمية بروتوكول الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام CSMA/CD . افترض أنه لاتوجد عقد نشطة أخرى، هل يمكن أن يصطدم الإرسالان المعادان من A و B؟ يكفينا هنا أخذ هذا المثال بعين الاعتبار: افترض أن A وB يبدآن الإرسال عند الوقت t = صفر فترة بت. سيكتشف كلٌّ منهما حدوث اصطدام عند t = 225 فترة بت، وسينتهيا من إرسال إشارة التشويش عند $t = 225 + 48$ = 273 فترة بت. افترض أن $K_A = 0$ و$K_B = 1$. في أي وقت ستُجدول العقدة B لإرسالها

المعاد؟ في أي وقت ستبدأ A الإرسال؟ (ملاحظة: ينبغي على العقدتين الانتظار إلى أن تصبح القناة خالية بعد العودة إلى الخطوة 2 – انظر البروتوكول). في أي وقت تصل إشارة A إلى B؟ هل ستُحجم B عن الإرسال في وقت الإرسال الذي جدولته؟

17. خذ في الاعتبار إيثرنت 100BASE-T سرعتها 100 ميجابت/ثانية، حيث كل العقد عليها موصّلة إلى مجمِّع (hub). للحصول على كفاءة مقدارها 0.50، كم ينبغي أن تكون المسافة القصوى بين أيٍّ من العقد والمجمِّع ؟ افترض إطاراً طوله 64 بايتاً ولا توجد مكرِّرات. هل تضمن تلك المسافة القصوى أيضاً أن العقدة المرسِلة A يتسنى لها اكتشاف ما إذا كان هناك عقدة تقوم بالإرسال بينما A ترسل؟ علل إجابتك. كيف تبدو المسافة القصوى التي حسبتَها مقارنةً بتلك المسافة التي يحددها معيار 100 ميجابت/ثانية فعلاً.

18. في هذا التمرين سوف نشتق تعبيراً رياضياً لكفاءة بروتوكول للوصول المتعدد يشبه بروتوكول CSMA/CD. في ذلك البروتوكول يُقسّم الوقت إلى شرائح ويتم تزامن كل موائمات الشبكة مع الشرائح. غير أنه بخلاف بروتوكول ألوها الشرائحي، يكون طول الشريحة هنا أقل بكثير من زمن الإطار (أي الزمن اللازم لإرسال إطار). دع S تمثل طول فترة الشريحة. افترض أن كل الإطارات لها طول ثابت وقدره $L = kRS$، حيث R هو معدل الإرسال على القناة و k رقم صحيح كبير. افترض أن هناك N عقدة، لدى كلٍّ منها عدد لانهائي من الإطارات تود إرسالها. سنفترض أيضاً أن $d_{prop} < S$، بحيث تتمكّن كل العقد من اكتشاف وجود اصطدام قبل نهاية مدة الإطار. يمكن وصف البروتوكول كالتالي:

- في كل شريحة، إذا لم تكن هناك عقدة تستحوذ على القناة، تقوم كل العقد بمحاولة استخدام القناة للإرسال، وتقوم كل عقدة بالإرسال أثناء الشريحة باحتمال p. إذا قامت عقدة واحدة بالضبط بالإرسال في تلك الشريحة فإنها تستحوذ على القناة طوال الشرائح الـ (k-1) التالية لكي ترسل إطارها بأكمله.
- إذا كانت هناك عقدة تستحوذ على القناة، فستُحجم كل العقد عن الإرسال إلى أن تنتهي العقدة التي تستحوذ على القناة من إرسال إطارها. بمجرد إرسال تلك القناة لإطارها، تقوم كل العقد بمحاولة استخدام القناة للإرسال.

لاحظ أن القناة تراوح مابين حالتين: الحالة المنتجة، والتي تستمر لمدة k شريحة بالضبط، والحالة غير المنتجة، والتي تستمر لعدد عشوائي من الشرائح. واضح أن

كفاءة القناة هي النسبة $k/(k+x)$، حيث x هو العدد المتوقع للشرائح المتتابعة غير المنتجة.

a. لقيم ثابتة لـ N و p، أوجد كفاءة هذا البروتوكول.

b. لقيمة ثابتة لـ N، أوجد قيمة p التي تحقق الحد الأقصى للكفاءة.

c. مستخدماً قيمة p التي حصلت عليها في الخطوة (b) أعلاه (والتي هي دالة في N)، أوجد قيمة الكفاءة عندما تقارب N مالانهاية.

d. اثبت أن الكفاءة تقارب 1 عندما يصبح طول الإطار كبيراً.

19. افترض أن عقدتين A وB موصلتان على طرفي كبل طوله 900 متر وأنه لدى كلٍّ منها إطار طوله 1000 بت (بما في ذلك كل التراويس والديباجات) تريد إرساله إلى الأخرى. تحاول كلا العقدتين الإرسال عند $t = 0$. افترض وجود 4 مكرّرات بين A وB يتسبب كلٌّ منها في تأخير يكافيء 20 بتاً. افترض أن معدل الإرسال هو 10 ميجابت/ثانية وأننا نستخدم بروتوكول CSMA/CD بفترة تراجع قدرها 512 بتاً. بعد أول تصادم، تسحب A القيمة $K = 0$ بينما تسحب ب القيمة $K = 1$ تبعاً لبروتوكول التراجع الأسي. اهمل إشارة التشويش والتأخير لفترة 96 بتاً.

a. ما هو تأخير الانتقال في اتجاه واحد بالثانية بين العقدتين A وB (بما في ذلك التأخير في المكرّرات).

b. في أي وقت (بالثانية) يتم تسليم الرزمة بأكملها من A إلى B؟

c. افترض الآن أن أ فقط لديها رزمة للإرسال وأن المكرّرات تم استبدالها بمحوّلات. افترض أن كل محول يتضمن تأخير معالجة قدره 20 بتاً بالإضافة إلى تأخير للتخزين والإرسال. في أي وقت (بالثانية) في هذه الحالة يتم تسليم الرزمة من A إلى B؟

20. خذ في الاعتبار الشكل 5-38 في تمرين 12. عيّن عناوين الماك وعناوين IP للواجهات على المضيف A، وكلا الموجّهين، والمضيف F. افترض أن المضيف A يرسل وحدة بيانات IP إلى المضيف F. أوجد عناوين ماك للمصدر والوجهة في الإطار الذي يغلف وحدة بيانات IP تلك عند: 1. إرسال الإطار من A إلى الموجّه على اليسار. 2. إرسال الإطار من الموجّه على اليسار إلى الموجّه على اليمين. 3. إرسال الإطار من الموجّه على اليمين إلى المضيف F. أوجد أيضاً عناوين IP للمصدر والوجهة في وحدة بيانات IP التي يغلفها ذلك الإطار في كل جزء من الأجزاء الثلاثة أعلاه من الرحلة.

21. افترض الآن أننا استبدلنا الموجّه على أقصى اليسار في الشكل 5-38 بمحوّل ووُصلت به كلٌّ من المضيفات A, B, C, D وكذلك الموجّه الأيمن على شكل نجمة. أوجد عناوين

ماك للمصدر والوجهة في الإطار الذي يغلف وحدة بيانات IP عند: 1. إرسال الإطار من A إلى الموجّه على اليسار. 2. إرسال الإطار من الموجّه على اليسار إلى الموجّه على اليمين. 3. إرسال الإطار من الموجّه على اليمين إلى المضيف F. أوجد أيضاً عناوين IP للمصدر والوجهة في وحدة بيانات IP التي يغلفها ذلك الإطار في كل جزء من الأجزاء الثلاثة أعلاه من الرحلة.

22. خذ في الاعتبار الشكل 5-26. افترض أن كل الوصلات تعمل بمعدل إرسال قدره 100 ميجابت/ثانية. ماهي الطاقة الإنتاجية الكلية للإرسال التي يمكن تحصيلها بين الأنظمة الطرفية الأربعة عشر في تلك الشبكة؟ ولماذا؟

23. افترض أن مفاتيح الأقسام الثلاثة في الشكل 5-26 تم استبدالها بمجمِّعات. كل الوصلات تعمل بمعدل إرسال قدره 100 ميجابت/ثانية. ماهي الطاقة الإنتاجية الكلية للإرسال التي يمكن تحصيلها بين الأنظمة الطرفية الأربعة عشر في تلك الشبكة؟ ولماذا؟

24. افترض أن كل المفاتيح في الشكل 5-26 تم استبدالها بمجمِّعات. كل الوصلات تعمل بمعدل إرسال قدره 100 ميجابت/ثانية. ماهي الطاقة الإنتاجية الكلية للإرسال التي يمكن تحصيلها بين الأنظمة الطرفية الأربعة عشر في تلك الشبكة؟ ولماذا؟

25. لنأخذ في الاعتبار طريقة عمل المحوّل المتعلم في سياق الشكل 5-24. افترض أن: (1) A ترسل إطاراً إلى D، (2) D ترد على A بإرسال إطار، (3) C ترسل إطاراً إلى D، (4) D ترد على C بإرسال إطار. افترض أن جدول المحوّل يكون خالياً في البداية. بيّن حالة جدول المحول قبل وبعد كلٍّ من تلك الخطوات الأربعة. لكل خطوة قم بتحديد الوصلات التي سيتم تمرير الإطار المرسل عليها، وعلل إجابتك باختصار.

26. تذكّر أن شبكات ATM تستخدم رزماً طولها 53 بايتاً تتألف من ترويسة طولها 5 بايتات وحمل آجر طوله 48 بايتاً. تعتبر 53 بايتاً قليلة بشكلٍ ملحوظ للرزم ثابتة الطول؛ فمعظم بروتوكولات الشبكات (مثل بروتوكول الإنترنت، والإيثرنت، وترحيل الإطارات، إلخ) تستخدم في المتوسط أطوالاً أكبر بكثير. أحد عيوب استخدام طول صغير للرزمة هو ضياع جزء كبير من سعة الإرسال (الحيز الترددي) للوصلة في إرسال بايتات العبء الإضافي؛ ففي هذه الحالة "تُهدر" 10٪ تقريباً من سعة الإرسال في إرسال ترويسة ATM. في هذا التمرين سنبحث في السبب وراء اختيار مثل هذا الطول القصير للرزمة. لهذا الغرض، افترض أن خلية ATM تتألف من L بايتاً (قد تختلف عن 48 بايتاً) وترويسة طولها 5 بايتات.

a. خذ في الاعتبار إرسال بيانات رقمية مكوّدة من مصدر صوت مباشرة على ATM. افترض أن المصدر يتم تكويده بمعدل 64 كيلوبت/ثانية. افترض أن كل خلية يتم ملؤها تماماً قبل أن يقوم المصدر بإرسالها إلى الشبكة. إن الوقت اللازم لملء الخلية هو تأخير الترزيم. احصل على تعبير رياضي لتأخير الترزيم (بالميللي ثانية) بدلالة L.

b. إذا زاد تأخير الترزيم عن 20 ميللي ثانية فقد يتسبب في حدوث صدى ملحوظ ومزعج. احسب تأخير الترزيم لـ $L = 1500$ (أي ما يناظر تقريباً أقصى طول ممكن لرزم الإيثرنت) وكذلك $L = 48$ (أي خلية ATM).

c. احسب تأخير "التخزين والإرسال" عند مفتاح ATM لوصلة معدل إرسالها R قيمته 155 ميجابت/ثانية (وتلك سرعة وصلة مفضلة في شبكات ATM) لكلٍ من $L = 1500$ و$L = 48$.

d. علّق على ميزات استخدام خلية قصيرة.

27. خذ في الاعتبار شبكة MPLS المبينة في الشكل 5-37، وافترض أن الموجّهين R5 و R6 مزوّدان بإمكانيات للتعامل مع MPLS. افترض أننا نود استخدام هندسة حركة المرور بحيث يتم تحويل الرزم الخارجة من R6 قاصدةً A إلى A عبر R6-R4-R3-R1، وتحويل الرزم الصادرة من R5 قاصدةً A إلى A عبر R5-R4-R2-R1. وضّح جداول MPLS الموجودة في الموجّهين R5 و R6، وكذلك الجدول المعدّل في R4، واللازمة لتحقيق ذلك.

28. خذ في الاعتبار مرة أخرى نفس السيناريو في التمرين السابق، ولكن افترض أن الرزم الخارجة من R6 قاصدةً D يتم تحويلها عبر R6-R4-R3 ، بينما الرزم من R5 قاصدةً D يتم تحويلها عبر R4-R2-R1-R3. وضّح جداول MPLS الموجودة في كل الموجّهات واللازمة لتحقيق ذلك.

❖ أسئلة مناقشة

نحثك على تصفح الإنترنت بحثاً عن إجابات للأسئلة التالية:

1. ماهو المدى التقريبي لسعر: موائم شبكة إيثرنت بسرعة 10/100 ميجابت/ثانية؟ موائم شبكة جيجابت إيثرنت؟. قارن هذه الأسعار بأسعار مودم هاتف بسرعة 56 كيلوبت/ثانية؟ أو بمودم ADSL؟

2. يتم تسعير المحوّل عادةً بناءً على عدد الواجهات التي يتضمنها (والتي تُعرف بالمنافذ في مصطلحات شبكة الإيثرنت). ما هو مدى السعر التقريبي لكل واجهة لمحوّل يتضمن فقط واجهات بسرعة 100 ميجابت/ثانية.

3. يمكن القيام بالعديد من مهام موائم الشبكة بواسطة برامجيات تعمل على المعالج المركزي للعقدة. ماهي مزايا وعيوب نقل تلك المهام من موائم الشبكة إلى العقدة؟
4. ابحث في الويب عن أرقام البروتوكولات المستخدمة في إطار إيثرنت لوحدة بيانات IP و رزمة ARP.
5. اقرأ المراجع [Xiao 2000; Huang 2002; RFC 3346] حول هندسة حركة مرور البيانات باستخدام MPLS. اسرد أهداف هندسة حركة مرور البيانات. أيٌّ من هذه الأهداف يمكن تحقيقه فقط باستخدام MPLS وأيّها يمكن تحقيقه ببروتوكولات أخرى موجودة غير MPLS؟ في الحالة الثانية، ما هي المزايا التي يوفرها استخدام MPLS؟

❖ مختبر إيثريل

ستجد على موقع الويب المصاحب لهذا الكتاب (http://www.aw1.com/kurose-ross) مختبر إيثيريل لاستكشاف طريقة عمل بروتوكول IEEE 802.3 وصيغة إطار الإيثرنت.

[**3Com 2007**] 3Com Corporation, "White paper: Understanding IP addressing: Everything you ever wanted to know," http://www.3com.com/other/pdfs/infra/corpinfo/en_US/501302.pdf

[**3GPP 2007**] Third Generation Partnership Project, http://www.3gpp.org/

[**802.11 Security 2007**] The Unofficial 802.11 Security Web Page, http://www.drizzle.com/~aboba/IEEE/

[**Abitz 1993**] P. Albitz and C. Liu, *DNS and BIND,* O'Reilly & Associates, Petaluma, CA, 1993.

[**Abramson 1970**] N. Abramson, "The Aloha System—Another Alternative for Computer Communications," *Proceedings of Fall Joint Computer Conference, AFIPS Conference,* p. 37, 1970.

[**Abramson 1985**] N. Abramson, "Development of the Alohanet," *IEEE Transactions on Information Theory,* Vol. IT-31, No. 3 (Mar. 1985), pp. 119–123.

[**Adler 2002**] M. Adler, "Tradeoffs in Probabilistic Packet Marking for IP Traceback*," Proceedings of 34th ACM Symposium on Theory of Computing (STOC)*, May 2002. http://www.cs.umass.edu/~micah/pubs/traceback.ps

[**Adya 2004**] A. Adya, W. J. Bolosky, M. Castro, G. Cermak, R. Chaiken, J. R. Douceur, J. Howell, J. R. Lorch, M. Theimer, R. P. Wattenhofer, "FARSITE: Federated, Available, and Reliable Storage for an Incompletely Trusted Environment." *Proceedings of the 5th OSDI*, December 2002. http://research.microsoft.com/~adya/pubs/osdi2002.pdf

[**Ahn 1995**] J. S. Ahn, P. B. Danzig, Z. Liu, and Y. Yan, "Experience with TCP Vegas: Emulation and Experiment," *Proceedings of ACM SIGCOMM '95* (Boston, MA, Aug. 1995), pp. 185–195. http://www.acm.org/sigcomm/sigcomm95/papers/ahn.html

[**Akamai 2007**] Akamai homepage, http://www.akamai.com

[**Akella 2003**] A. Akela, S. Seshan, A. Shaikh, "An Empirical Evaluation of Wide-Area Internet Bottlenecks," *Proc. 2003 ACM Internet Measurement Conf.* (Miami FL, Nov. 2003).

[**Alvestrand 1997**] H. Alvestrand, "Object Identifier Registry," http://www.alvestrand.no/harald/objectid/top.html.

[**Anderson 1995**] J. B. Andersen, T. S. Rappaport, S. Yoshida, "Propagation Measurements and Models for Wireless Communications Channels," *IEEE Communications Magazine*, (Jan. 1995), pp. 42-49.

[**Appenzeller 2004**] G. Appenzeller, I. Kelassy, N. McKeown, "Sizing Router Buffers," *Proc. 2004 ACM SIGCOMM* (Portlan, OR, Aug. 2004).

[**Aprisma 2007**] Aprisma homepage, http://www.aprisma.com/

[**ARIN 1996**] ARIN, "IP allocation report," ftp://rs.arin.net/netinfo/ip_network_allocations

[**Ash 1998**] G. R. Ash, *Dynamic Routing in Telecommunications Networks,* McGraw Hill, NY, NY, 1998.

[**ASO-ICANN 2007**] The Address Supporting Organization home page, http://www.aso.icann.org

[**AT&T SLM 2006**] AT&T Business, "AT&T Enterprise Hosting Services Service Guide," http://www.att.com/abs/serviceguide/docs/eh_sg.pdf

[**Atheros 2006**] Atheros Communications Inc. "Atheros AR5006 WLAN Chipset Product Bulletins," http://www.atheros.com/pt/AR5006Bulletins.htm

[**ATM Forum 2007**] The ATM Forum Web site, http://www.atmforum.com/

[**Ayanoglu 1995**] E. Ayanoglu, S. Paul, T. F. La Porta, K. K. Sabnani, R. D. Gitlin, "AIR-MAIL: A Link-Layer Protocol for Wireless Networks," *ACM ACM/Baltzer Wireless Networks Journal*, 1: 47–60, February 1995. http://www.bell-labs.com/user/sanjoy/airmail.ps.Z

[**Bakre 1995**] A. Bakre, B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," *Proceedings of the 15th International Conf. on Distributed Computing Systems (ICDCS)*, May 1995, pp. 136–143. ftp://paul.rutgers.edu/pub/badri/itcp-tr314.ps.Z

[**Balakrishnan 1995**] H. Balakrishnan, S. Seshan, R. H. Katz, "Improving Reliable Transport and Handoff Performance in Cellular Wireless Networks," *ACM Wireless Networks*, 1, no. 4 (December 1995). http://nms.lcs.mit.edu/~hari/papers/winet.ps

[**Balakrishnan 1997**] H. Balakrishnan, V. Padmanabhan, S. Seshan, R. Katz, "A Comparison of Mechanisms for Improving TCP Performance over Wireless Links," *IEEE/ACM Transactions on Networking* 5, no. 6 (December 1997). http://nms.lcs.mit.edu/~hari/papers/ton.ps

[**Baptista 2003**] A. Baptista, T. Leen, Y. Zhang, A. Chawla, D. Maier, W. Feng, W. Feng, J. Walpole, C. Silva, J. Freire, "Environmental Observation and Forecasting Systems: Vision, Challenges and Successes of a Prototype," *Encyclopedia of Physical Science and Technology* (R. A. Meyers, Ed.), Academic Press, Third Edition, Vol. 5., pp 565-581.

[**Baran 1964**] P. Baran, "On Distributed Communication Networks," *IEEE Transactions on Communication Systems,* Mar. 1964. Rand Corporation Technical report with the same title (Memorandum RM-3420-PR, 1964). http://www.rand.org/publications/RM/RM3420/

[**Bardwell 2007**] J. Bardwell, "You Believe You Understand What You Think I Said … The Truth About 802.11 Signal And Noise Metrics: A Discussion Cclasifying Often-Misused 802.11 WLAN Terminologies," http://madwifi.org/attachment/wiki/UserDocs/RSSI/you_believe_D100201.pdf?format=raw

[**Baset 2006**] S. A. Baset and H. Schulzrinne, "An analysis of the Skype peer-to-peer Internet Telephony Protocol," *Proc. 2006 IEEE Infocom* (Barcelona, Spain, Apr. 2006).

[**BBC 2001**] BBC news online "A Small Slice of Design," April 2001, http://news.bbc.co.uk/1/low/sci/tech/1264205.stm

[**BBC Multicast 2007**] BBC, "BBC Multicast Trial," http://support.bbc.co.uk/multicast

[**Bender 2000**] P. Bender, P. Black, M. Grob, R. Padovai, N. Sindhushayana, A. Viterbi, "CDMA/HDR: A bandwidth-efficient high-speed wireless data service for nomadic users," *IEEE Commun. Mag.*, Vol. 38, No. 7 (July 200) pp. 70-77.

[**Berners-Lee 1989**] T. Berners-Lee, CERN, "Information Management: A Proposal," Mar. 1989, May 1990. http://www.w3.org/History/1989/proposal.html

[**Berners-Lee 1994**] T. Berners-Lee, R. Cailliau, A. Luotonen, H. Frystyk Nielsen, and A.Secret, "The World-Wide Web," *Communications of the ACM,* Vol. 37, No. 8 (Aug. 1994), Pages 76–82

[**Bernstein 2007**] D. Bernstein, "SYN Cookies," http://cr.yp.to/syncookies.html

[**Bertsekas 1991**] D. Bertsekas and R. Gallagher, *Data Networks, 2nd Ed.,* Prentice Hall, Englewood Cliffs, NJ, 1991.

[**Bhagwat 2003**] P. Bhagwat, B. Raman, D. Sanghi, "Turning 802.11 Inside Out," *Proceedings of the 2003 ACM Hotnets II Workshop*, Cambridge, MA (November 2003). http://nms.lcs.mit.edu/HotNets-II/papers/inside-out.pdf

[**Bhimani 1996**] Anish Bhimani: "Securing the Commercial Internet," *Communications of the ACM,* Vol. 39 No. 6: 29–35; March 1996

[**Biddle 2003**] P. Biddle, P. England, M. Peinado, B. Willman, "The Darknet and the Future of Content Distribution." 2002 ACM Workshop on Digital Rights Management, (Nov. 2002, Washington, D.C.) http://crypto.stanford.edu/DRM2002/darknet5.doc

[**Biersack 1992**] E. W. Biersack, "Performance evaluation of forward error correction in ATM networks," *Proceedings of ACM SIGCOMM'92* (Baltimore, MD 1992), pp. 248–257. http://www.acm.org/pubs/articles/proceedings/comm/144179/p248-biersack/p248-biersack.pdf

[**BIND 2004**] Internet Software Consortium page on BIND, http://www.isc.org/bind.html

[**Bisdikian 2001**] C. Bisdikian, "An Overview of the Bluetooth Wireless Technology," *IEEE Communications Magazine*, No. 12 (December 2001): 86–9 4.

[**Bishop 2003**] M. Bishop, *Computer Security: Art and Science,* Boston: Addison Wesley, Boston MA, 2003

[**BitTorrent 2007**] BitTorrent.org homepage, http://www.bittirrent.org

[**Black 1995**] U. Black, *ATM Volume I: Foundation for Broadband Networks*, Prentice Hall, 1995.

[**Black 1997**] U. Black, ATM, Volume II: Signaling in Broadband Networks, Prentice Hall, 1997.

[**Blaze 1996**] M. Blaze,W. Diffie, R. Rivest, B. Schneier, T. Shimomura, E. Thompson, and M. Weiner, "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security," http://www.counterpane.com/keylength.html

[**Bluetooth 2002**] R. Morrow, *Bluetooth: Operation and Use*, New York: McGraw-Hill, 2002.

[**Blumenthal 2001**] M. Blumenthal, D. Clark, "Rethinking the Design of the Internet: The End-to-end Arguments vs. the Brave New World," *ACM Transactions on Internet Technology*, Vol. 1, No. 1, (August 2001) pp. 70-109.

[**Bochman 1984**] G. V. Bochmann and C. A. Sunshine, "Formal methods in communication protocol design," *IEEE Transactions on Communications,* Vol. COM-28, No. 4 (Apr. 1980), pp. 624–631.

[**Bolot 1994**] J-C. Bolot and T. Turletti, "A rate control scheme for packet video in the Internet," *Proceedings of IEEE Infocom,* 1994, pp. 1216–1223. ftp://ftp-sop.inria.fr/rodeo/bolot/94.Video_control.ps.gz

[**Bolot 1996**] J-C. Bolot and Andreas Vega-Garcia, "Control Mechanisms for Packet Audio in the Internet," *Proceedings of IEEE Infocom,* 1996, pp. 232–239. ftp://ftp-sop.inria.fr/rodeo/bolot/96.Audio_ctl.ps.gz

[**Boutremans 2002**] C. Boutremans, G. Iannaccone, C. Diot, "Impact of Link Failures on VoIP Performance," *12th International Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV),* Miami, May 2002. http://ipmon.sprint.com/pubs_trs/pubs/gianluca/voip.pdf

[**Bradner 1996**] S. Bradner, A. Mankin, *IPng: Internet Protocol Next Generation,* Addison-Wesley, Reading, MA, 1996.

[**Brakmo 1995**] L. Brakmo and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," *IEEE Journal of Selected Areas in Communications,* Vol. 13, No. 8, pp. 1465–1480, Oct. 1995. ftp://ftp.cs.arizona.edu/xkernel/Papers/jsac.ps.Z

[**Breslau 2000**] L. Breshlau, E. Knightly, S. Shenker, I. Stoica, H. Zhang, "Endpoint Admission Control: Architectural Issues and Performance," *Proc. 2000 ACM SIGCOMM* (Stockholm, Sweden, Aug. 2000)

[**Brodnik 1997**] A. Brodnik, S. Carlsson, M. Degemark, S. Pink, "Small Forwarding Tables for Fast Routing Lookups," *Proceedings of ACM SIGCOMM '97* (Cannes, France, Oct. 1997), pp. 3–15. http://www.acm.org/sigs/sigcomm/sigcomm97/papers/p192.html

[**Brown 1997**] K. Brown, S. Singh, "M-TCP: TCP for Mobile Cellular Networks," *ACM CCR* 27, no. 5 (1997). http://www.cs.pdx.edu/~singh/ftp/mtcp.ps.gz.

[**Bryant 1988**] B. Bryant, "Designing an Authentication System: A Dialogue in Four Scenes," http://web.mit.edu/kerberos/www/dialogue.html

[**Bush 1945**] V. Bush, "As We May Think," *The Atlantic Monthly,* July 1945. http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm

[**Byers 1998**] J. Byers, M. Luby, M. Mitzenmacher, A Rege, "A digital fountain approach to reliable distribution of bulk data," *Proceedings of ACM SIGCOMM '98* (Vancouver, 1998, Aug. 1998), pp. 56–67. http://www.acm.org/sigcomm/sigcomm98/tp/abs_05.html

[**Cablelabs 2007**] CableLabs homepage, http://www.cablelabs.com

[**CacheLogic 2007**] CacheLogic homepage, http://www.cachelogic.com

[**Caesar 2005**] M. Caesar, J. Rexford, "BGP Routing Policies in ISP Networks," *IEEE Network Magazine*, vol. 19, no. 6 (Nov. 2005).

[**Caldwell 2007**] C. Caldwell, "The Prime Pages," http://www.utm.edu/research/primes/prove

[**Cardwell 2000**] N. Cardwell, S. Savage, T. Anderson, "Modeling TCP Latency," *Proceedings of the 2000 IEEE Infocom Conference*, (Tel-Aviv, Israel), March, 2000. http://www.cs.ucsd.edu/users/savage/papers/Infocom2000tcp.ps

[**CASA 2007**] Center for Collaborative Adaptive Sensing of the Atmosphere, http://www.casa.umass.edu

[**Casner 1992**] Casner, S., Deering, S., "First IETF Internet Audiocast," *ACM SIGCOMM Computer Communications Review,* Vol. 22, No. 3 (July 1992), pp. 92–97. http://citeseer.nj.nec.com/casner92first.html

[**Ceiva 2007**] Ceiva homepage, http://www.ceiva.com/

[**CENS 2007**] Center for Embedded Network Sensing, http://www.cens.ucla.edu/

[**Cerf 1974**] V. Cerf and R. Kahn, "A Protocol for Packet Network Interconnection," *IEEE Transactions on Communications Technology,* Vol. COM-22, No. 5, pp. 627–641.

[**CERT 1999-04**] CERT, "Advisory CA-1999-04: Melissa Macro Virus," http://www.cert.org/advisories/CA-1999-04.html

[**CERT 2001-09**] CERT, "Advisory 2001-09: Statistical Weaknesses in TCP/IP Initial Sequence Numbers," http://www.cert.org/advisories/CA-2001-09.html

[**CERT 2001-19**] CERT, "Advisory CA-2001-19: "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL," http://www.cert.org/advisories/CA-2001-19.html

[**CERT 2003-04**] CERT, “CERT Advisory CA-2003-04 MS-SQL Server Worm,” http://www.cert.org/advisories/CA-2003-04.html

[**CERT 2003-04**] CERT, “CERT Advisory CA-2003-04 MS-SQL Server Worm,” http://www.cert.org/advisories/CA-2003-04.html

[**CERT 2007**] CERT Coordination Center, http://www.cert.org/advisories

[**CERT Filtering 2007**] CERT, “Packet Filtering for Firewall Systems,” http://www.cert.org/tech_tips/packet_filtering.html

[**CERT Smurf 1998**] CERT(r) Advisory CA-98.01, “smurf IP Denial-of-Service Attacks,” http://www.cert.org/advisories/CA-1998-01.html

[**CERT SYN 1996**] CERT, “Advisory CA-96.21: TCP SYN Flooding and IP Spoofing Attacks,” http://www.cert.org/advisories/CA-1998-01.html

[**CERT 2004 Summaries**] CERT, “CERT Summaries,” http://www.cert.org/summaries/

[**Chao 2001**] H. J. Chao, C. Lam, E. Oki, Broadband Packet Switching Technologies—A Practical Guide to ATM Switches and IP Routers, John Wiley & Sons, 2001.

[**Chapman 1992**] B. Chapman, “Network (In)Security Through Packet Filtering,” *Third UNIX Security Symposium, sponsored by USENIX Association*, (Baltimore, MD), 1992, http://www.greatcircle.com/pkt_filtering.html

[**Checkpoint 2004**] Checkpoint Web site, http://www.checkpoint.com.

[**Chen 2000**] G. Chen, D. Kotz, “A Survey of Context-Aware Mobile Computing Research,” *Technical Report TR2000-381*, Dept. of Computer Science, Dartmouth College, November, 2000. http://www.cs.dartmouth.edu/~dfk/papers/chen:survey-tr.pdf

[**Chen 2006**] K.-T. Chen, C.-Y. Huang, P. Huang, C.-L. Lei, “Quantifying Skype User Satisfaction,” *Proc. 2006 ACM SIGCOMM* (Pisa, Italy, Sept. 2006).

[**Cheswick 2000**] Bill Cheswick, Hal Burch, Steve Branigan, “Mapping and Visualizing the Internet,” *Proc. 2000 Usenix Conference* (June 2000, San Diego) http://www.usenix.org/publications/library/proceedings/usenix2000/general/full_papers/cheswick /cheswick_html/ mapping.html

[**Chiu 1989**] D. Chiu and R. Jain, “Analysis of the Increase and Decrease Algorithms for Congestion Avoidance in Computer Networks,” *Computer Networks and ISDN Systems,* Vol. 17, No. 1, pp. 1–14. http://www.cis.ohio-state.edu/~jain/papers/cong_av.htm

[**Christiansen 2001**] M. Christiansen, K. Jeffay, D. Ott, F. D. Smith, “Tuning Red for Web Traffic,” *IEEE/ACM Transactions on Networking*, Vol. 9, No. 3 (June 2001), pp. 249–264, http://www.cs.unc.edu/~jeffay/papers/IEEE-ToN-01.pdf

[**Chu 2000**] Y Chu, S. Rao, H. Zhang, “The Case for End System Multicast,” *Proceedings of ACM SIGMETRICS 2000,* (Santa Clara, CA, Aug. 2000). http://www.cs.cmu.edu/~sanjay/Papers/sigmetrics-2000.ps.gz

[**Chuang 2005**] S. Chuang, S. Iyer, N. McKeown, “Practical Algorithms for Performance Guarantees in Buffered Crossbars,” *Proc. 2005 IEEE Infocom*.

[**Cicconetti 2006**] C. Cicconetti, L. Lenzini, A. Mingozi, K. Eklund, “Quality of Service Support in 802.16 Networks,” *IEEE Network Magazine*, Mar./Apr. 2006, pp. 50-55.

[**Cisco 12000 2007**] Cisco Systems, "Cisco 12000 Series Gigabit Switch Routers," http://www.cisco.com/univercd/cc/td/doc/pcat/12000.htm

[**Cisco 8500 2007**] Cisco Systems Inc., "Catalyst 8500 Campus Switch Router Architecture," http://www.cisco.com/univercd/cc/td/doc/product/l3sw/8540/rel_12_0/w5_6f/softcnfg/1cfg8500.pdf

[**Cisco CiscoWorks 2000**] Cisco Systems, Cisco Works2000 homepage, http://www.cisco.com/warp/public/cc/pd/wr2k/index.shtml

[**Cisco NAT 2007**] Cisco Systems Inc, "How NAT Works," http://www.cisco.com/warp/public/556/nat-cisco.shtml

[**Cisco NAPA 2007**] Cisco Systems Inc., "Cisco Network Application Performance Analysis (NAPA) Solution," http://www.cisco.com/en/US/products/sw/netmgtsw/index.html

[**Cisco QoS 2007**] Cisco Systems Inc, "Advanced QoS Services for the Intelligent Internet," http://www.cisco.com/warp/public/cc/pd/iosw/ioft/ioqo/tech/qos_wp.htm

[**Cisco Queue 2007**] Cisco Systems Inc., "Interface Queue Management," http://www.cisco.com/warp/public/614/16.html

[**Cisco Security 2007**] Cisco Systems Inc., "Why You Need a Firewall," http://www.cisco.com/en/US/products/sw/secursw/ps743/products_user_guide_chapter09186a008007f303.html

[**Cisco Switches 2007**] Cisco Systems Inc., "Cisco Catalyst 1900/2820 - Affordable Switching Solutions" http://www.cisco.com/warp/public/cc/pd/si/index.shtml

[**Cisco SYN 2007**] Cisco Systems Inc., "Defining Strategies to Protect Against TCP SYN Dential of Service Attacks," http://www.cisco.com/warp/public/707/4.html#tcpsyn

[**CISN 2004**] California Integrated Seismic Network, http://www.cisn.org/

[**Claffy 1998**] K. Claffy, G. Miller, and K. Thompson, "The Nature of the Beast: Recent Traffic Measurements from an Internet Backbone," *Proceedings of Inet '98,* (Geneva, Switzerland, July 1998), http://www.caida.org/outreach/papers/1998/Inet98/

[**Clark 1988**] D. Clark, " The Design Philosophy of the DARPA Internet Protocols, *Proceedings of ACM SIGCOMM'88,* (Stanford, CA), Aug. 1988, Vol. 18, No. 4, http://www.acm.org/sigcomm/ccr/archive/1995/jan95/ccr-9501-clark.html.

[**Clarke 2002**] I. Clarke, T. W. Hong, S. G. Miller, O. Sandberg, B. Wiley, "Protecting Free Expression Online with Freenet," *IEEE Internet Computing*, January–February 2002, pp. 40–49. http://freenet.sourceforge.net/papers/freenet-ieee.pdf

[**Cnet 2000**] Cnet news.com, "Leading Web Sites Under Attack," http://news.com.com/2100-1017-236683.html

[**Cohen 1977**] D. Cohen, "Issues in Transnet Packetized Voice Communication," *Proceedings of the Fifth Data Communications Symposium*, (Snowbird, Utah, September 1977) pp. 6-13.

[**Cookie Central 2007**] Cookie Central homepage, http://www.cookiecentral.com

[**CoolStreaming 2005**] X. Zhang, J. Liu, B. Li, Y. Yum, "CoolStreaming/DONet: a data-driven overlay network for peer-to-peer live media streaming," *Proc. IEEE Infocom*, (March 2005, Miami FL).

[**Cormen 2001**] T. H. Cormen, *Introduction to Algorithms, 2nd Ed.,* MIT Press, Cambridge, MA, 2001.

[**Crow 1997**] B. Crow, I. Widjaja, J. Kim, P. Sakai, “IEEE 802.11 Wireless Local Area Networks,” *IEEE Communications Magazine,* Sept. 1997, pp. 116–126.

[**Crowcroft 1995**] J. Crowcroft, Z. Wang, A. Smith, J. Adams, *“A Comparison of the IETF and ATM Service Models,” IEEE Communications Magazine,* Nov./ Dec. 1995, pp. 12–16. http://citeseer.nj.nec.com/crowcroft95rough.html

[**Crowcroft 1999**] J. Crowcroft, M. Handley, and I. Wakeman, *Internetworking Multimedia,* Morgan-Kaufman, San Francisco, 1999.

[**Culler 2004**] D. Culler, D. Estrin, M. Srivastava, “Overview of Sensor Networks,” *IEEE Computer*, Vol. 37, No. 8, pp. 41-49, Aug. 2004.

[**Cusumano 1998**] M.A. Cusumano and D.B. Yoffie, *Competing on Internet Time: Lessons from Netscape and its Battle with Microsoft,* Free Press, NY, NY, 1998

[**Daemen 2000**] J. Daemen, V. Rijmen, “The Block Cipher Rijndael,” in *Smart Card Research and Applications, LNCS 1820*, (J. J. Quisquater, B. Schneier, eds.), Springer-Verlag, 2000, pp. 288–296.

[**Daigle 1991**] J. N. Daigle, *Queuing Theory for Telecommunications,* Addison-Wesley, Reading, MA, 1991.

[**Dalal 1978**] Y. Dalal, R. Metcalfe, “Reverse Path Forwarding of Broadcast Packets,” *Communications of the AC* M, Vol. 21, No. 12, (Dec. 1978), pp. 1040-1048.

[**Davie 2000**] B. Davie and Y. Rekhter, *MPLS: Technology and Applications*, Morgan Kaufmann Series on Networking, 2000.

[**Davies 2004**] G. Davies, M. Hardt and F. Kelly, “Network dimensioning, service costing and pricing in a packet switched environment,” *Telecommunications Policy*, Vol. 28, pp. 391-412, 2004.

[**Danielyan 2001**] E. Danielyan, “Goodbye DES, Welcome AES,” *Internet Protocol Journal* 4(2), June 2001. http://www.cisco.com/en/US/about/ac123/ac147/ac174/about_cisco_ipj_archive_issues_list.html

[**DEC 1990**] Digital Equipment Corporation, “In Memoriam: J. C. R. Licklider 1915–1990,” SRC Research Report 61, Aug. 1990. http://www.memex.org/licklider.pdf

[**DeClercq 2002**] J. DeClercq, O. Paridaens, “Scalability Implications of Virtual Private Networks,” *IEEE Communications Magazin*e, 40(5), May 2002, pp. 151–157.

[**Deering 1990**] S. Deering, D. Cheriton, “Multicast routing in datagram internetworks and extended LANs,” *ACM Transactions on Computer Systems,* Vol. 8, No. 2 (1990), pp. 85–110.

[**Deering 1996**] S. Deering, D. Estrin, D. Faranacci, V. Jacobson, C. Liu, L. Wei, “The PIM Architecture for Wide Area Multicasting,” *IEEE/ACM Transactions on Networking,* Vol. 4, No. 2 (Apr. 1996), pp. 153–162.

[**Demers 1990**] A. Demers, S. Keshav, and S. Shenker, “Analysis and Simulation of a Fair Queuing Algorithm,” *Internetworking: Research and Experience,* Vol. 1, No. 1, pp. 3–26, 1990..

[**Denning 1997**] D. Denning (Editor), P. Denning (Preface), *Internet Besieged: Countering Cyberspace Scofflaws,* Addison-Wesley, Reading, MA, 1997.

[**dhc 2007**] IETF Dynamic Host Configuration working group, http://www.ietf.org/html.charters/dhc-charter.html

[**Dialpad 2004**] Dialpad homepage, http://www.dialpad.com

[**Diffie 1976**] W. Diffie and M. E. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory,* Vol IT-22 (1976), pp. 644–654.

[**Diffie 1998**] W. Diffie and S. Landau, *Privacy on the Line, The Politics of Wiretapping and Encryption,* MIT Press, Cambridge MA, 1998.

[**Digital Signature 2004**] Digital Signature Trust Company, http://www.trustdst.com/

[**Diggavi 2004**] S. N. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. R. Calderbank, "Great Expectations: The Value of Spatial Diversity in Wireless Networks," *Proceedings of the IEEE*,vol. 92, no. 2, pp. 217-270, Feb. 2004.

[**Diot 2000**] C. Diot, B. N. Levine, B. Lyles, H. Kassem, D. Balensiefen, "Deployment Issues for the IP Multicast Service and Architecture," *IEEE Network,* Vol. 14, No. 1 (Jan./Feb. 2000), pp. 78–88, http://signl.cs.umass.edu/pubs/brian.ieeenetwork00.ps.gz

[**Dodge 2007**] M. Dodge, "An Atlas of Cyberspaces," http://www.cybergeography.org/atlas/isp_maps.html

[**Donahoo 2000**] M. Donahoo, K. Calvert, *TCP/IP Sockets in C: Practical Guide for Programmers*, Morgan Kaufman, 2000.

[**Dornan 2001**] A. Dornan, *The Essential Guide to Wireless Communications Applications: From Cellular Systems to WAP and M-Commerce*, Prentice Hall, Upper Saddle River, N.J., 2001.

[**Doucer 2002**] J. R. Douceur, "The Sybil Attack," *Proc. of the IPTPS'02 Workshop*, (Cambridge, MA, Mar. 2002).

[**Droms 1999**] R. Droms, T. Lemon, *The DHCP Handbook*, Macmillan Technical Publishing, Indianapolis, IN, 1999.

[**DSL 2007**] DSL Forum, http://www.dslforum.org/

[**EFF 1999**] Electronic Frontier Foundation, "Frequently Asked Questions (FAQ) About the Electronic Frontier Foundation's DES Cracker Machine," http://www.eff.org/pub/Privacy/Crypto/Crypto_misc/DESCracker/HTML/19980716_eff_des_faq.html

[**Elgamal 2001**] A. Elgamal, F. Seible, F. Vernon, M. Trivedi, M. Fraser, "On-Line Structural Monitoring and Data Management," *Proceedings, 6th Seismic Research Workshop,* California Department of Transportation*,* Sacramento, California, June 12–13, 2001. http://www.calit2.net/eci/caltrans_health_monitoring_paper.pdf

[**Ellis 1987**] H. Ellis, "The Story of Non-Secret Encryption," http://www.cesg.gov.uk/site/publications/media/ellis.pdf

[**Ericsson 2007**] Ericsson, "EDGE: Introduction of High-Speed Data in GSM/GPRS Networks." http://www.ericsson.com/products/white_papers_pdf/edge_wp_technical.pdf

[**Estrin 1997**] D. Estrin, M. Handley, A. Helmy, P. Huang, D. Thaler, "A Dynamic Bootstrap Mechanism for Rendezvous-based Multicast Routing," *Proceedings of IEEE Infocom '98,* (New York, NY, April 1998). http://ceng.usc.edu/~helmy/infocom-bootstrap-99.pdf

[**Estrin 1998b**] Deborah Estrin, V. Jacobson, D. Farinacci, L. Wei, Steve Deering, Mark Handley, David Thaler, Ching-Gung Liu, Puneet Sharma, A. Helmy, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Motivation and Architecture," work in progress, http://netweb.usc.edu/pim/pimsm/PIM-Arch.ps.gz.

[**Estrin 2002**] D. Estrin, D. Culler, K. Pister, "Connecting the Physical World with Pervasive Networks," *IEEE Pervasive Computing*, 1,1 (Jan.–March 2002).

[**Ethereal 2007**] Ethereal homepage, http://www.ethereal.com

[**Faloutsos 1999**] C. Faloutsos, M. Faloutsos, P. Faloutsos, "What Does the Internet Look Like? Empirical Laws of the Internet Topology," *Proceedings of ACM SIGCOMM 1999*, Boston, MA, September 1999.

[**Feamster 2004**] N. Feamster, J. Winick, J. Rexford, "A Model for BGP Routing for Network Engineering," *Proceedings of 2004 ACM Sigmetrics,* NY, NY (June 2004). http://www.research.att.com/~jrex/papers/whatifatron.pdf

[**Feldmeier 1988**] D. Feldmeier, "Improving Gateway Performance with a Routing Table Cache," *Proc. 1988 IEEE Infocom Conference* (New Orleans LA, Mar. 1988).

[**Feldmeier 1995**] D. Feldmeier, "Fast Software Implementation of Error Detection Codes," *IEEE/ACM Transactions on Networking,* Vol. 3., No. 6 (Dec. 1995), pp. 640–652.

[**FIPS 1995**] Federal Information Processing Standard, "Secure Hash Standard," FIPS Publication 180-1. http://www.itl.nist.gov/fipspubs/fip180-1.htm

[**FIPS-46-1 1988**] US National Bureau of Standards, "Data Encryption Standard," Federal Information Processing Standard (FIPS) Publication 46-1, Jan. 1988. http://www.itl.nist.gov/fipspubs/fip46-2.htm

[**Fletcher 1982**] J. G. Fletcher, "An Arithmetic Checksum for Serial Transmissions," *IEEE Transactions on Communications,* Vol. 30, No. 1 (Jan. 1982), pp. 247–253.

[**Floyd 1999**] S. Floyd and K. Fall, "Promoting the Use of End-to-End Congestion Control in the Internet," *IEEE/ACM Transactions on Networking,* Vol. 6, No. 5 (Oct. 1998), pp. 458–472. http://www.icir.org/floyd/end2end-paper.html

[**Floyd 2000**] S. Floyd, M. Handley, J. Padhye, J. Widmer, "Equation-Based Congestion Control for Unicast Applications," Proceedings 2000 ACM Sigcomm Conference, (Stockholm, Sweden, Aug. 2000). http://www.icir.org/tfrc/tcp-friendly.pdf

[**Floyd 2001**] S. Floyd, "A Report on Some Recent Developments in TCP Congestion Control," *IEEE Communications Magazine* (April 2001), http://www.aciri.org/floyd/papers/report_Jan01.pdf

[**Floyd 2007**] S. Floyd, "References on RED (Random Early Detection) Queue Management," http://www.icir.org/floyd/red.html

[**Floyd Synchronization 1994**] S. Floyd, V. Jacobson, "Synchronization of Periodic Routing Messages," *IEEE/ACM Transactions on Networking,* Vol. 2, No. 2 (Apr. 1997), pp. 122–136. http://www.aciri.org/floyd/papers/sync_94.ps.Z

[**Floyd TCP 1994**] S. Floyd, "TCP and Explicit Congestion Notification," *ACM Computer Communication Review,* Vol. 24, No. 5, pp. 10–23, Oct. 1994. http://www.aciri.org/floyd/papers/tcp_ecn.4.ps.Z

[**Fluhrer 2001**] S. Fluhrer, I. Mantin, A. Shamir, "Weaknesses in the Key Scheduling Algorithm of RC4," *Eighth Annual Workshop on Selected Areas in Cryptography*, Toronto, August 2002. http://www.drizzle.com/~aboba/IEEE/rc4_ksaproc.pdf

[**Fortz 2000**] B. Fortz, M. Thorup, "Internet Traffic Engineering by Optimizing OSPF Weights," *Proceedings of 2000 IEEE Infocom*. http://www.ieee-infocom.org/2000/papers/165.ps.

[**Fortz 2002**] B. Fortz, J. Rexford, M. Thorup, "Traffic Engineering with Traditional IP Routing Protocols," *IEEE Communication Magazine,* October 2002. http://www.research.att.com/~jrex/papers/ieeecomm02.ps

[**Foster 2002**] I. Foster, "The Grid: A New Infrastructure for 21st Century Science," *Physics Today*, 55(2):42–47, 2002, http://www.aip.org/pt/vol-55/iss-2/p42.html.

[**Freephone 2004**] "Freephone: Why use the Plain Old Telephone when you can get so much better on the Internet?" http://www-sop.inria.fr/rodeo/fphone/

[**Friedman 1999**] T. Friedman, D. Towsley "Multicast Session Membership Size Estimation," *Proc. IEEE Infocom '99* (New York, USA, March 1999) ftp://gaia.cs.umass.edu/pub/Friedman99_Infocom99.ps.gz

[**Frost 1994**] J. Frost, "BSD Sockets: A Quick and Dirty Primer," http://world.std.com/~jimf/papers/sockets/sockets.html

[**Gallager 1983**] R. G. Gallager, P. A. Humblet, P. M. Spira, "A Distributed Algorithm for Minimum Weight-Spanning Trees," *ACM Trans. on Programming Languages and Systems,* 1(5), (January 1983), pp. 66–77.

[**Gao 2001**] L. Gao, J. Rexford, "Stable Internet Routing Without Global Coordination," *IEEE/ACM Trans. Networking,* 9(6), pp. 681–692, December 2001. http://www.research.att.com/~jrex/papers/sigmetrics00.long.pdf

[**Garces-Erce 2003**] L. Garces-Erce, K. W. Ross, E. Biersack, P. Felber, G. Urvoy-Keller, "TOPLUS: Topology Centric Lookup Service," *Fifth International Workshop on Networked Group Communications (NGC'03*), Munich, September 2003. http://cis.poly.edu/~ross/papers/TOPLUS.pdf

[**Gartner 2003**] F. C. Gartner, "A Survey of Self-Stabilizing Spanning-Tree Construction Algorithms," *Technical Report IC/2003/38*, Swiss Federal Institute of Technology (EPFL), School of Computer and Communication Sciences, June 10, 2003. http://ic2.epfl.ch/publications/documents/IC_TECH_REPORT_200338.pdf.

[**Gauthier 1999**] L. Gauthier, C. Diot, and J. Kurose, "End-to-end Transmission Control Mechanisms for Multiparty Interactive Applications on the Internet," *Proceedings of IEEE Infocom '99,* (New York, NY, Apr. 1999). ftp://ftp.sprintlabs.com/diot/infocom99-mimaze.zip

[**Giacopelli 1990**] J. Giacopelli, M. Littlewood, W. D. Sincoskie "Sunshine: A high performance self-routing broadband packet switch architecture," *1990 International Switching Symposium.* An extended version of this paper appeared in *IEEE J. Sel. Areas in Common*., Vol. 9, No. 8 (Oct. 1991), pp. 1289–1298.

[**Girard 1990**] A. Girard, *Routing and Dimensioning in Circuit-Switched Networks,* Addison-Wesley, Reading, MA, 1990.

[**Glitho 1995**] R. Glitho and S. Hayes (eds.), special issue on Telecommunications Management Network, *IEEE Communications Magazine,* Vol. 33, No. 3 (Mar. 1995).

[**Glitho 1998**] R. Glitho, "Contrasting OSI Systems Management to SNMP and TMN," *Journal of Network and Systems Management,* Vol. 6, No. 2 (June 1998), pp. 113–131.

[**Gnutella 2007**] "The Gnutella Protocol Specification, v0.4"
http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf

[**Goodman 1997**] David J. Goodman, *Wireless Personal Communications Systems*, Prentice-Hall, 1997.

[**Goodman 1997b**] D. Goodman (Chair), *The Evolution of Untethered Communications,* National Academy Press, Washington DC, Dec. 1997.
http://www.nap.edu/readingroom/books/evolution/index.html

[**Goralski 1999**] W. Goralski, *Frame Relay for High-Speed Networks,* John Wiley, New York, 1999.

[**Goralski 2001**] W. Goralski, *Optical Networking and WDM,* Osborne/McGraw-Hill, Berkeley, CA, 2001.

[**Griffin 2002**] T. Griffin, "Interdomain Routing Links,"
http://www.research.att.com/~griffin/interdomain.html

[**Gummadi 2003**] K. P. Gummadi, R. J. Dunn, S. Saroiu, S. D. Gribble, H. M. Levy, J. Zahorjan, "Measurement, Modeling, and Analysis of a Peer-to-Peer File-Sharing Workload," *Proceedings of the 19th ACM Symposium on Operating Systems Principles (SOSP-19),* October 2003.
http://www.cs.washington.edu/homes/tzoompy/publications/sosp/2003/abstract.html

[**Gupta 1998**] P. Gupta, S. Lin, N. McKeown. "Routing lookups in hardware at memory access speeds," *Proc. IEEE Infocom 1998* (San Francisco, CA, April 1998), pp. 1241–1248. http://tiny-tera.stanford.edu/~nickm/papers/Infocom98_lookup.pdf

[**Gupta 2001**] P. Gupta, N. McKeown, "Algorithms for Packet Classification," *IEEE Network Magazine,* Vol. 15, No. 2 (Mar./Apr. 2001), pp. 24–32,
http://klamath.stanford.edu/~pankaj/paps/ieeenetwork_tut_01.pdf

[**Halabi 2000**] S. Halabi, *Internet Routing Architectures, 2nd Ed* ., Cisco Press, 2000.

[**Hamada 1997**] T. Hamada, H. Kamata, S. Hogg, "An Overview of the TINA Management Architecture," *Journal of Network and Systems Management,* Vol. 5. No. 4 (Dec. 1997). pp. 411–435.

[**Heidemann 1997**] J. Heidemann, K. Obraczka, and J. Touch, "Modeling the Performance of HTTP over Several Transport Protocols," *IEEE/ACM Transactions on Networking,* Vol. 5, No. 5 (Oct. 1997), pp. 616–630. http://www.isi.edu/~johnh/PAPERS/Heidemann96a.html

[**Held 2001**] G. Held, *Data Over Wireless Networks: Bluetooth, WAP, and Wireless LANs*, McGraw-Hill, 2001.

[**Hersent 2000**] O. Hersent, D. Gurle, J-P Petit, *IP Telephony: Packet-Based Multimedia Communication Systems*," Pearson Education Limited, Edinburgh, 2000.

[**Hinden 2007**] R. Hinden, "IP Next Generation (IPng),"
http://playground.sun.com/pub/ipng/html/ipng-main.html

[**Holbrook 1999**] H. Holbrook, D. Cheriton, "IP Multicast Channels: EXPRESS Support for Large-Scale Single-Source Applications," *Proceedings of ACM SIGCOMM '99* (Boston, MA, Aug. 1999). http://www.acm.org/sigs/sigcomm/sigcomm99/papers/session2-3.html

[**Hollot 2002**] C.V. Hollot, V. Misra, D. Towsley, W. Gong, "Analysis and design of controllers for AQM routers supporting TCP flows," IEEE Transactions on Automatic Control, Vol. 47, No. 6 (June 2002), pp. 945-959. http://www1.cs.columbia.edu/~misra/pubs/TAC_special.pdf

[**Huang 2002**] C. Haung, V. Sharma, K. Owens, V. Makam, "Building Reliable MPLS Networks Using a Path Protection Mechanism," *IEEE Communications Magazine,* 40(3), March 2002, pp. 156–162.

[**Huitema 1998**] C. Huitema, *IPv6: The New Internet Protocol, 2nd Ed.,* Prentice Hall, Englewood Cliffs, NJ, 1998.

[**Huston 1999a**] G. Huston, "Interconnection, Peering, and Settlements—Part I," *The Internet Protocol Journal,* Vol. 2, No. 1, (March 1999). http://www.cisco.com/warp/public/759/ipj_2-1/ipj_2-1_ps1.html

[**Huston 1999b**] G. Huston, "Interconnecting, Peering, and Settlements—Part II," *The Internet Protocol Journal,* Vol. 2, No. 2 (June 1999). http://www.cisco.com/warp/public/759/ipj_2-2/ipj_2-2_ps1.html

[**Huston 2001**] G. Huston, "Analyzing the Internet BGP Routing Table," *The Internet Protocol Journal*, Vol. 4, No. 1 (Mar. 2001), http://www.cisco.com/warp/public/759/ipj_4-1/ipj_4-1_bgp.html

[**IAB 2007**] Internet Architecture Board, http://www.iab.org/iab/

[**IANA 2007**] Internet Assigned Number Authority homepage, http://www.iana.org/

[**ICANN 2007**] The Internet Corporation for Assigned Names and Numbers, http://www.icann.org

[**IEC Optical 2007**] IEC Online Education, "Optical Access," http://www.iec.org/online/tutorials/opt_acc/

[**IEEE 802 2007**] "IEEE 802 LAN/MAN Standards Committee," http://www.ieee802.org/

[**IEEE 802.11 1999**] IEEE 802.11, 1999 Edition (ISO/IEC 8802-11: 1999) IEEE Standards for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Network—Specific Requirements—Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, http://standards.ieee.org/getieee802/download/802.11-1999.pdf

[**IEEE 802.15 2007**] IEEE 802.15 Working Group for WPAN. http://grouper.ieee.org/groups/802/15/.

[**IEEE 802.1X**] IEEE Std 802.1X-2001 Port-Based Network Access Control, http://standards.ieee.org/reading/ieee/std_public/description/lanman/802.1x-2001_desc.html

[**IETF 2007**] Internet Engineering Task Force homepage, http://www.ietf.org

[**IETF dnsext 2004**] IETF DNS Extensions Working Group, http://www.ietf.org/html.charters/dnsext-charter.html

[**Interlinknetworks 2004**] Interlinknetworks, "Introduction to 802.1x for Wireless Local Area Networks," http://www.interlinknetworks.com/resource/wp5-1-1.htm

[**IMAP 2007**] The IMAP Connection, http://www.imap.org/

[**Interlinknetworks 2004**] Internlinknetworks, "Introduction to 802.1x for Wireless Local Area Networks," http://www.interlinknetworks.com/resource/wp5-1-1.htm

[**Ioannidis 2000**] S. Ioannidis, A. Keromytis, S. Bellovin, J. M. Smith, “Implementing a Distributed Firewall,” *Proceedings of the ACM Computer and Communications Security (CCS)* 2000, (Athens, Greece), pp. 190–199, http://www.cis.upenn.edu/~strongman/Papers/df.pdf

[**Iren 1999**] S. Iren, P. Amer, P. Conrad, “The Transport Layer: Tutorial and Survey,” *ACM Computing Surveys,* Vol 31, No 4, (Dec 1999). http://www.cis.udel.edu/~amer/PEL/survey/

[**ISC 2007**] Internet Systems Consortium, http://www.isc.org.

[**ISO 1987**] International Organization for Standardization, “Information processing systems — Open Systems Interconnection—,”International Standard 8824 (Dec. 1987). http://asn1.elibel.tm.fr/en/standards/index.htm

[**ISO X.680** 1998] International Organization for Standardization, “X.680: ITU-T Recommendation X.680 (1997) | ISO/IEC 8824-1:1998, Information Technology—Abstract Syntax Notation One (ASN.1): Specification of Basic Notation.” http://asn1.elibel.tm.fr/en/standards/index.htm

[**ITU 2000**] International Telecommunication Union, “Recommendation X.509 (11/93) Information technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks”
http://www.itu.int/rec/recommendation.asp?type=items&lang=e&parent=T-REC-X.509-200003-I

[**ITU 2007**] The ITU Web site, http://www.itu.int/

[ITU Statistics 2007] International Telecommunication Union, “ICT Statistics,” http://www.itu.int/ITU-D/icteye/Reports.aspx

[**ITU-T Q.2931 1994**] “Broadband Integrated Service Digital Network (B-ISDN) Digital Subscriber Signaling System no.2 (DSS2) User Network Interface Layer 3 Specification for Basic Call/Connection Control,” *ITU-T Recommendation Q.2931*, Geneva: International Telecommunication Union, 1994.

[**Iyer 2002**] S. Iyer, R. Zhang, N. McKeown, “Routers with a Single Stage of Buffering,” *Proceedings 2002 ACM Sigcomm Conference,*
http://www.acm.org/sigs/sigcomm/sigcomm2002/papers/routersingle.pdf.

[**Jacobson 1988**] V. Jacobson, “Congestion Avoidance and Control,” *Proceedings of ACM SIGCOMM ‘88,* (Stanford, CA, Aug. 1988), pp. 314–329, ftp://ftp.ee.lbl.gov/papers/congavoid.ps.Z

[**Jain 1989**] R. Jain, “A Delay-Based Approach for Congestion Avoidance in Interconnected Heterogeneous Computer Networks,” *ACM Computer Communications Review,* Vol. 19, No. 5 (1989), pp. 56–71. http://www.cis.ohio-state.edu/~jain/papers/delay.htm

[**Jain 1994**] R. Jain, *FDDI Handbook: High-Speed Networking Using Fiber and Other Media,* Addison-Wesley, Reading, MA, 1994.

[**Jain 1996**] R. Jain. S. Kalyanaraman, S. Fahmy, R. Goyal, and S. Kim, “Tutorial Paper on ABR Source Behavior,” *ATM Forum*/96-1270, Oct. 1996. http://www.cis.ohio-state.edu/~jain/atmf/a96-1270.htm

[**Jaiswal 2003**] S. Jaiswal, G. Iannaccone, C. Diot, J. Kurose, D. Towsley, “Measurement and Classification of Out-of-Sequence Packets in a Tier-1 IP backbone,” *Proceedings of 2003 INFOCO* M, ftp://gaia.cs.umass.edu/pub/Jaiswal03_oos.pdf.

[**Jakobson 1993**] G. Jacobson and M. Weissman, “Alarm Correlation,” *IEEE Network Magazine,* 1993, pp. 52–59.

[**Ji 2003**] P. Ji, Z. Ge, J. Kurose, D. Towsley, "A Comparison of Hard-state and Soft-state Signaling Protocols," *Proceedings of 2003 ACM SIGCOMM*, http://www.acm.org/sigs/sigcomm/sigcomm2003/papers/p251-ji.pdf

[**Jiang 2001**] W. Jiang, J. Lennox, H. Schulzrinne, K. Singh, "Towards Junking the PBX: Deploying IP Telephony," *NOSSDAV'01* (Port Jefferson, NY, June 2001). http://www.cs.columbia.edu/~hgs/papers/Jian0106_Junking.pdf.

[**Jimenez 1997**] D. Jimenez, "Outside Hackers Infiltrate MIT Network, Compromise Security," *The Tech,* Vol. 117, No. 49 (Oct. 1997), p. 1. http://www-tech.mit.edu/V117/N49/hackers.49n.html

[**Jin 2004**] C. Jin, D. X. We, S. Low, "FAST TCP: Motivation, architecture, algorithms, performance," *Proc. IEEE Infocom*, Hong Kong, March 2004, http://netlab.caltech.edu/pub/papers/FAST-csreport2003.pdf.

[**Kaaranen 2001**] H. Kaaranen, S. Naghian, L. Laitinen, A. Ahtiainen, Valtteri Niemi*, UMTS Networks, Architecture, Mobility and Service* s, John Wiley &Sons, 2001.

[**Kahn 1967**] D. Kahn, *The Codebreakers, the Story of Secret Writing,* The Macmillan Company, 1967.

[**Kahn 1978**] R. E. Kahn, S. Gronemeyer, J. Burchfiel, R. Kunzelman, "Advances in Packet Radio Technology," *Proc. of the IEEE*, 66, 11 (November 1978).

[**Kangasharju 2000**] J. Kangasharju, K. W. Ross, and J. W. Roberts, "Performance Evaluation of Redirection Schemes in Content Distribution Networks ," *Proceedings of 5th Web Caching and Content Distribution Workshop, Lisbon*, Portugal, May 2000, Lisbon, Portugal. http://www.terena.nl/conf/wcw/Proceedings/S4/S4-2.ps

[**Kapoor 1997**] H. Kapoor, "CoreBuilder 5000 Switch Module Architecture," 3 Corporation, white paper, number 500645.

[**Kar 2000**] K. Kar, M. Kodialam, T. V. Lakshman, "Minimum Interference Routing of Bandwidth Guaranteed Tunnels with MPLS Traffic Engineering Applications," *IEEE J. Selected Areas in Communications*, December, 2000. http://www.bell-labs.com/org/11347A/paper/minint_jsac.pdf

[**Karol 1987**] M. Karol, M. Hluchyj, A. Morgan, "Input Versus Output Queuing on a Space-Division Packet Switch," *IEEE Transactions on Communications,* Vol. COM-35, No. 12 (Dec. 1987), pp. 1347–1356.

[**Katzela 1995**] I. Katzela, and M. Schwartz. "Schemes for Fault Identification in Communication Networks," *IEEE/ACM Transactions on Networking,* Vol. 3, No. 6 (Dec. 1995), pp. 753–764.

[**Kaufman 1995**] C. Kaufman, R. Perlman, M. Speciner, *Network Security, Private Communication in a Public World,* Prentice Hall, Englewood Cliffs, NJ, 1995.

[**KaZaA 2004**] KaZaA homepage, http://www.kazaa.com

[**Kelly 2003**] T. Kelly, *Scalable TCP: Improving Performance in Highspeed Wide Area Networks,* http://www-lce.eng.cam.ac.uk/~ctk21/papers/scalable_improve_hswan.pdf.

[**Kende 2000**] M. Kende, "The Digital Handshake: Connecting Internet Backbones," FCC Report, 2000, http://www.fcc.gov/Bureaus/OPP/working_papers/oppwp32.pdf

[**Keshav 1998**] S. Keshav, R. Sharma, "Issues and Trends in Router Design," *IEEE Communications Magazine,* Vol. 36, No. 5 (May 1998), pp. 144–151.

[**Kilkki 1999**] K. Kilkki, *Differentiated Services for the Internet*, Macmillan Technical Publishing, Indianapolis, IN, 1999.

[**Kleinrock 1961**] L. Kleinrock, "Information Flow in Large Communication Networks," RLE Quarterly Progress Report, July 1961.

[**Kleinrock 1964**] L. Kleinrock, *1964 Communication Nets: Stochastic Message Flow and Delay,* McGraw-Hill, NY, NY, 1964.

[**Kleinrock 1975**] L. Kleinrock, *Queuing Systems, Vol. 1,* John Wiley, New York, 1975.

[**Kleinrock 1975b**] L. Kleinrock and F. A. Tobagi, "Packet Switching in Radio Channels: Part I—Carrier Sense Multiple-Access Modes and Their Throughput-Delay Characteristics," *IEEE Transactions on Communications,* Vol. COM-23, No. 12 (Dec. 1975), pp. 1400–1416.

[**Kleinrock 1976**] L. Kleinrock, *Queuing Systems, Vol. 2,* John Wiley, New York, 1976.

[**Kleinrock 2004**] L. Kleinrock, "The Birth of the Internet," http://www.lk.cs.ucla.edu/LK/Inet/birth.html

[**Kohler 2004**] E. Kohler, M. Handley, S. Floyd, J. Padhye, DCCP homepage, http://www.icir.org/kohler/dccp/

[**Korhonen 2003**] J. Korhonen, *Introduction to 3G Mobile Communications*, 2nd ed., Artech House, 2003.

[**Krishnamurthy 2001**] B. Krishnamurthy, and J. Rexford, *Web Protocols and Practice: HTTP/1.1, Networking Protocols, and Traffic Measurement*, Addison-Wesley, Boston, MA, 2001.

[**Kurose 1996**] J. F. Kurose, Unix Network Programming, http://manic.cs.umass.edu/~amldemo/courseware/intro.html

[**Labovitz 1997**] C. Labovitz, G. R. Malan, F. Jahanian, "Internet Routing Instability," *Proceedings of ACM SIGCOMM '97* (Cannes, France, 1997), Pages 115–126. http://www.acm.org/sigcomm/sigcomm97/papers/p109.html

[**Labrador 1999**] M. Labrador, S. Banerjee, "Packet Dropping Policies for ATM and IP Networks," *IEEE Communications Surveys*, Vol. 2, No. 3 (Third Quarter 1999), pp. 2–14, http://www.comsoc.org/livepubs/surveys/public/3q99issue/banerjee.html

[**Lakshman 1997**] T. V. Lakshman, U. Madhow, "The Performance of TCP/IP for Networks with High Bandwidth-Delay Products and Random Loss," *IEEE/ACM Transactions on Networking,* Vol. 5 No. 3 (1997). pp. 336–350. http://citeseer.ist.psu.edu/lakshman96performance.html

[**Lam 1980**] S. Lam, "A Carrier Sense Multiple Access Protocol for Local Networks," *Computer Networks,* Vol. 4 (1980), pp. 21–32, 1980.

[**Lamport 1981**] L. Lamport, "Password Authentication with Insecure Communication", *Communications of the ACM,* Vol. 24, No. 11 (Nov. 1981), pp. 770–772.

[**Larmouth 1996**] J. Larmouth, *Understanding OSI,* International Thomson Computer Press 1996. Chapter 8 of this book deals with ASN.1 and is available on-line at http://www.salford.ac.uk/iti/books/osi/all.html#head8

[**Larsen 1997**] A. Larsen, "Guaranteed Service: Monitoring Tools," *Data Communications,* June 1997, pp. 85–94.

[**Lawton 2001**] G. Lawton, "Is IPv6 Finally Gaining Ground?" *IEEE Computer Magazine* (Aug. 2001), pp. 11–15.

[**Leiner 1998**] B. Leiner, V. Cerf, D. Clark, R. Kahn, L. Kleinrock, D. Lynch, J. Postel, L. Roberts, and S. Woolf, "A Brief History of the Internet," http://www.isoc.org/internet/history/brief.html

[**Liang 2004**] J. Liang, R. Kumar, K.W. Ross, "Understanding KaZaA", http://cis.poly.edu/~ross/papers/.

[**Lin 2001**] Y. Lin, I. Chlamtac, *Wireless and Mobile Network Architectures*, John Wiley and Sons, New York, NY, 2001.

[**Liu 2002**] B. Liu, D. Goeckel, D. Towsley, "TCP-Cognizant Adaptive Forward Error Correction in Wireless Networks," *Proceedings of Globe Internet 2002.* ftp://gaia.cs.umass.edu/pub/wirelessTCPtech.pdf

[**Luotonen 1998**] A. Luotonen, *Web Proxy Servers,* Prentice Hall, Englewood Cliffs, New Jersey, 1998.

[**Lynch 1993**] D. Lynch, M. Rose, *Internet System Handbook,* Addison-Wesley, Reading, MA, 1993.

[**Macedonia 1994**] Macedonia, M. R., Brutzman, D. P., "MBone Provides Audio and Video Across the Internet," *IEEE Computer Magazine,* Vol. 27, No. 4 (Apr. 1994), pp. 30–36. ftp://taurus.cs.nps.navy.mil/pub/mbmg/mbone.html

[**Maconachy 2001**] W.V. Maconachy, C. Schou, D. Ragsdale, D. Welch, "A Model for Information Assurance: an Integrated Approach," *Proceedings of the 2001 IEEE Workshop on Information Assurance and Security*, (West Point, NY), 2001, http://www.itoc.usma.edu/Documents/Workshop2001/paperW2C3(55).pdf

[**Maennel 2002**] O. Maennel, A. Feldmann, "Realistic BGP Traffic for Test Labs," *Proceedings. of 2002 ACM Sigcomm*, http://www.acm.org/sigs/sigcomm/sigcomm2002/papers/bgplab.pdf.

[**Mahdavi 1997**] J. Mahdavi and S. Floyd, "TCP-Friendly Unicast Rate-Based Flow Control," unpublished note, Jan. 1997. http://www.psc.edu/networking/papers/tcp_friendly.html

[**Mainwaring 2002**] A.Mainwaring, R. Szewczyk, D. Culler, J. Anderson "Wireless Sensor Networks for Habitat Monitoring" *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA*), 2002. http://citeseer.ist.psu.edu/mainwaring02wireless.html

[**Manelli 2001**] T. Manelli, "What Happened to Internet Appliances?" *PC World,* (April 2001), http://www.pcworld.com/news/article/0,aid,47184,00.asp

[**manet 2007**] IETF Mobile Ad-hoc Networks (manet) Working Group, http://www.ietf.org/html.charters/manet-charter.html

[**Maymounkov 2002**] P. Maymounkov and D. Mazières. "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric." *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02*), pp. 53–65, March 2002.

[**McAuley 1994**] A. McAuley, "Weighted Sum Codes for Error Detection and Their Comparison with Existing Codes," *IEEE/ACM Transactions on Networking,* Vol. 2, No. 1 (Feb. 1994), pp. 16–22.

[**McCumber 1991**] J. McCumber, "Information Systems Security: A Comprehensive Model," *Proceedings of the 14th National Computer Security Conference*, (Baltimore, MD), 1991.

[MCI 2004] MCI, "Terms and Conditions: Service Level Agreement," http://global.mci.com/terms/sla/

[**McKeown 1997a**] N. McKeown, M. Izzard, A. Mekkittikul,W. Ellersick, M. Horowitz, "The Tiny Tera: A Packet Switch Core," *IEEE Micro Magazine,* Jan.–Feb. 1997. http://tiny-tera.stanford.edu/~nickm/papers/HOTI_96.ps.

[**McKeown 1997b**] N. McKeown, "A Fast Switched Backplane for a Gigabit Switched Router," *Business Communications Review,* Vol. 27, No. 12. http://www.bcr.com/bcrmag/12/mckeown.htm

[**McKusik 1996**] M. K. McKusick, K. Bostic, M. Karels, and J. Quarterman, *The Design and Implementation of the 4.4BSD Operating System,* Addison-Wesley, Reading, MA, 1996.

[**McQuillan 1980**] J. McQuillan, I. Richer, E. Rosen, "The New Routing Algorithm for the Arpanet," *IEEE Transactions on Communications,* COM-28(5) (May 1980), pp. 711–719.

[**Medhi 1997**] D. Medhi and D. Tipper (eds.), Special Issue: Fault Management in Communication Networks, *Journal of Network and Systems Management,* Vol. 5. No. 2 (June 1997).

[**Metcalfe 1976**] R. M. Metcalfe and D. R. Boggs. "Ethernet: Distributed Packet Switching for Local Computer Networks," *Communications of the Association for Computing Machinery,* Vol. 19, No. 7, (July 1976), pp. 395–404. http://www.acm.org/classics/apr96/

[**Microsoft Player Media 2007**] Microsoft Windows Media homepage, http://www.microsoft.com/windows/windowsmedia/

[**Miller 1997**] M.A. Miller, *Managing Internetworks with SNMP,* 2nd ed., M & T Books, New York, 1997.

[**Mockapetris 1988**] P. V. Mockapetris, K. J. Dunlap, "Development of the Domain Name System," *Proceedings of SIGCOMM '88*, Stanford, CA, 1988. http://citeseer.nj.nec.com/mockapetris88development.html

[**Molinero-Fernandez 2002**] P. Molinaro-Fernandez, N. McKeown, H. Zhang, "Is IP Going to Take Over the World (of Communications)?" *Proc. 2002 ACM Hotnets*, http://www.acm.org/sigcomm/HotNets-I/papers/fernandez.pdf

[**Molle 1987**] M. L. Molle, K. Sohraby, and A. N. Venetsanopoulos, "Space-Time Models of Asynchronous CSMA Protocols for Local Area Networks," *IEEE Journal on Selected Areas in Communications,* Vol. 5, No. 6, (1987) pp. 956–968.

[**Molva 1999**] R. Molva, "Internet Security Architecture," Computer Networks and ISDN Systems, Vol. 31, No. 8 (1999), pp. 787-804.

[**Moore 2003**] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, N. Weaver, "Inside the Slammer Worm," http://www.caida.org/outreach/papers/2003/sapphire2/

[**Mouly 1992**] M. Mouly, M. Pautet, *The GSM System for Mobile Communications*, Cell and Sys, Palaiseau, France, 1992.

[**Moy 1998**] J. Moy, *OSPF: Anatomy of An Internet Routing Protocol*, Addison-Wesley, Reading, MA, 1998.

[**mrouted 1996**] "mrouted," v3.8 of DVMRP routing software for various workstation routing platforms, ftp://parcftp.xerox.com/pub/net-research/ipmulti

[**Mukherjee 1997**] B. Mukherjee, Optical Communication Networks, McGraw-Hill, 1997.

[**Murphy 2003**] S. Murphy, "BGP Security Vulnerabilities Analysis," draft-ietf-idr-bgp-vuln-00.txt, June 2003, ftp://ftp.rfc-editor.org/in-notes/internet-drafts/draft-ietf-idr-bgp-vuln-00.txt

[**Nahum 2002**] E. Nahum, T. Barzilai, D. Kandlur, "Performance Issues in WWWServers," *IEEE/ACM Transactions on Networking*, 10(1), February 2002, http://www.research.ibm.com/people/n/nahum/publications/ton02-www-camera.pdf.

[**Nesbitt 2002**] S. Nesbitt, "Network Appliances," Jan. 2002, About.com, http://netappliances.about.com/cs/settopboxes/.

[**Net2Phone 2004**] http://www.net2phone.com/

[**Netcraft 2004**] The Netcraft Web Server Survey, Netcraft Web Site, http://www.netcraft.com/survey/

[**Netscape Cookie 1999**] Netscape Communications Corp., "Persistent Client State http Cookies," http://home.netscape.com/newsref/std/cookie_spec.html

[**Netscape SSL 1998**] Netscape Communications Corps, "Introduction to SSL," http://developer.netscape.com/docs/manuals/security/sslin/

[**Neuman 1994**] B. Neuman and T. Tso, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communication Magazine,* Vol. 32, No. 9 (Sept. 1994), pp. 33–38.

[**Neumann 1997**] R. Neumann, "Internet Routing Black Hole," *The Risks Digest: Forum on Risks to the Public in Computers and Related Systems,* Vol. 19, No. 12 (May 1997). http://catless.ncl.ac.uk/Risks/19.12.html#subj1.1

[**Nielsen 1997**] H. F. Nielsen, J. Gettys, A. Baird-Smith, E. Prud'hommeaux, H. W. Lie, and C. Lilley, "Network Performance Effects of HTTP/1.1, CSS1, and PNG," *W3C Document,* 1997 (also appears in *Proceedings of ACM SIGCOMM '97,* Cannes, France, pp. 155–166). http://www.acm.org/sigcomm/sigcomm97/papers/p102.html

[**NIST 1993**] National Institute of Standards and Technology, "Federal Information. Data Encryption Standard," Processing Standards Publication 46-2, 1993. http://www.itl.nist.gov/fipspubs/fip46-2.htm

[**NIST 1999**] National Institute of Standards and Technology, "Data Encryption Standard Fact Sheet," http://csrc.nist.gov/cryptval/des/des.txt

[**NIST 1999b**] National Institute of Standards and Technology, "Draft Federal Information Processing Standard (FIPS) 46-3, Data Encryption Standard (DES), and Request for Comments," http://csrc.nist.gov/cryptval/des/fr990115.htm

[**NIST 2001**] National Institute of Standards and Technology, "Advanced Encryption Standard (AES)," Federal Information Processing Standards 197, Nov. 2001, http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[**Nmap 2007**] Nmap homepage, http://www.insecure.com/nmap

[**Nonnenmacher 1998**] J. Nonnenmacher, E. Biersak, D. Towsley, "Parity-Based Loss Recovery for Reliable Multicast Transmission," *IEEE/ACM Transactions on Networking,* Vol. 6, No. 4 (Aug. 1998), pp. 349–361. ftp://gaia.cs.umass.edu/pub/NBT97:fec.ps.gz

[**Nortel 2004**] Nortel Networks, Optivity Portfolio, http://www.nortelnetworks.com/products/01/optivity

[**NTIA 1998**] National Telecommunications and Information Administration (NTIA), U.S. Department of Commerce, "Management of Internet names and addresses," Docket Number: 980212036-8146-02. http://www.ntia.doc.gov/ntiahome/domainname/6_5_98dns.htm

[**Odlyzko 2003**] A. Odlyzko, "Internet Traffic Growth: Sources and Implications," A. M. Optical Transmission Systems and Equipment for WDM Networking II, *Proc. SPIE*,. 5247, 2003, pp. 1–15. http://www.dtc.umn.edu/~odlyzko/doc/itcom.internet.growth.pdf.

[**OpenView2007**] HP OpenView homepage, http://www.openview.hp.com/

[**Overpeer 2004**] Overpeer Inc., http://www.overpeer.com..

[**OSS 2007**] OSS Nokalva, "ASN.1 Resources," http://www.oss.com/asn1/

[**Padhye 2000**] J. Padhye, V. Firoiu, D. Towsley, J. Kurose, "Modeling TCP Reno Performance: A Simple Model and its Empirical Validation," *IEEE/ACM Transactions on Networking,* Vol. 8 No. 2 (April 2000). pp. 133–145.

[**Padhye 2001**] J. Padhye, S. Floyd, "On Inferring TCP Behavior," In *Proceedings of ACM SIGCOMM,* 2001, (San Diego, CA), 2001v. http://www.aciri.org/floyd/papers/tbit.pdf

[**Pan 1997**] P. Pan and H. Schulzrinne, "Staged Refresh Timers for RSVP," In 2nd Global Internet Conference, Phoenix, 1997. http://www.cs.columbia.edu/~pingpan/papers/timergi.pdf

[**Parekh 1993**] A. Parekh and R. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case," *IEEE/ACM Transactions on Networking,* Vol. 1, No. 3 (June 1993), pp. 344–357.

[**Partridge 1992**] C. Partridge, S. Pink, "An Implementation of the Revised Internet Stream Protocol (ST-2)," *Journal of Internetworking: Research and Experience* 3(1), March 1992. http://www.sics.se/cna/publications/ST-2.ps

[**Partridge 1998**] C. Partridge, et al. "A Fifty Gigabit per second IP Router," *IEEE/ACM Transactions on Networking,* Vol. 6, No. 3 (Jun. 1998), pp. 237–248.

[**Paxson 1997**] V. Paxson, "End-to-end Internet packet dynamics," *Proceedings of ACM SIGCOMM '97,* (Sept. 1997, Cannes, France). http://www.acm.org/sigcomm/sigcomm97/papers/p086.html

[**Perkins 1994**] A. Perkins, "Networking with Bob Metcalfe," *The Red Herring Magazine,* Nov. 1994. http://www.herring.com/mag/issue15/bob.html

[**Perkins 1998a**] C. Perkins, O. Hodson and V. Hardman, "A Survey of Packet Loss Recovery Techniques for Streaming Audio," *IEEE Network Magazine,* Sept./ Oct. 1998, pp. 40–47.

[**Perkins 1998b**] C. Perkins, *Mobile IP: Design Principles and Practice*, Addison-Wesley, Reading, MA, 1998.

[**Perkins 2000**] C. Perkins, *Ad Hoc Networking*, Addison-Wesley, Reading, MA, 2000.

[**Perlman 1999**] R. Perlman, *Interconnections: Bridges, Routers, Switches, and Internet-working Protocols,* 2nd ed., Addison-Wesley Professional Computing Series, Reading, MA, 1999.

[**PGPI 2007**] The International PGP Home Page, http://www.pgpi.org

[**Phifer 2000**] L. Phifer, "The Trouble with NA *T," The Internet Protocol Journal,* Vol. 3, No. 4 (Dec. 2000), http://www.cisco.com/warp/public/759/ipj_3-4/ipj_3-4_nat.html

[**Pickholtz 1982**] R. Pickholtz, D. Schilling, L. Milstein, "Theory of Spread Spectrum Communication—a Tutorial," *IEEE Transactions on Communications,* Vol. COM-30, No. 5 (May 1982), pp. 855–884.

[**Piscatello 1993**] D. Piscatello and A. Lyman Chapin, *Open Systems Networking,* Addison-Wesley, Reading, MA, 1993.

[**Point Topic 2006**] Point Topic Ltd., *World Broadband Statistics Q1 2006*, http://www.point-topic.com

[**QuickTime 2007**] QuickTime homepage, http://www.apple.com/quicktime

[**Quittner 1998**] J. Quittner, M. Slatalla, *Speeding the Net: The Inside Story of Netscape and How it Challenged Microsoft,* Atlantic Monthly Press, 1998.

[**Ramakrishnan 1990**] K. K. Ramakrishnan and Raj Jain, "A Binary Feedback Scheme for Congestion Avoidance in Computer Networks," *ACM Transactions on Computer Systems,* Vol. 8, No. 2 (May 1990), pp. 158–181.

[**Raman 1999**] S. Raman, S. McCanne, "A Model, Analysis, and Protocol Framework for Soft State-based Communication," *Proceedings of ACM SIGCOMM '99* (Boston, MA, Aug. 1999). http://www.acm.org/sigs/sigcomm/sigcomm99/papers/session1-2.html

[**Ramaswami 1998**] R. Ramaswami, K. Sivarajan, *Optical Networks: A Practical Perspective,* Morgan Kaufman Publishers, 1998

[**Ramjee 1994**] R. Ramjee, J. Kurose, D. Towsley, and H. Schulzrinne, "Adaptive Playout Mechanisms for Packetized Audio Applications in Wide-Area Networks," *Proceeding IEEE Infocom 94.* ftp://gaia.cs.umass.edu/pib/Ramj94:Adaptive.ps.Z

[**Rao 1996**] K. R. Rao and J. J. Hwang, *Techniques and Standards for Image, Video and Audio Coding,* Prentice Hall, Englewood Cliffs, NJ, 1996.

[**RAT 2007**] Robust Audio Tool, http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/

[**Ratnasamy 2001**] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A Scalable Content-Addressable Network," *In Proceedings of ACM SIGCOMM,* 2001, (San Diego, CA), 2001. http://www.acm.org/sigcomm/sigcomm2001/p13.html

[**RealNetworks 2007**] RealNetworks homepage, http://www.realnetworks.com

[**Reid 2003**] N. Reid and R. Seide, *802.11 (Wi-Fi) Networking Handbook*, McGraw-Hill/ Osborne, New York, 2003.

A note on Internet Request for Comments (RFCs): Copies of Internet RFCs are maintained at multiple sites. The RFC URLs below all point into the RFC archive at the Information Sciences Institute (ISI), maintained at the RFC Editor of the Internet Society (the body that oversees the RFCs). Other RFC sites include http://www.faqs.org/rfcs, http://www.pasteur.fr/other/computer/RFC (located in France), and http://www.csl.sony.co.jp/rfc/ (located in Japan). Internet RFCs can be updated or obsoleted by later RFCs. We encourage you to check the sites listed above for the most up-to-date information. The RFC search facility at ISI, http://www.rfc-editor.org/rfc.html, will allow you to search for an RFC and show updates to that RFC.

[**RFC 001**] S. Crocker, "Host Software," RFC 001 (the *very first* RFC!). http://www.rfc-editor.org/rfc/rfc1.txt

[**RFC 741**] D. Cohen, “Specifications for the Network Voice Protocol NVP”, RFC 741, Nov. 1977. ftp://ftp.rfc-editor.org/in-notes/rfc741.txt

[**RFC 768**] J. Postel, “User Datagram Protocol,” RFC 768, Aug. 1980. http://www.rfc-editor.org/rfc/rfc768.txt

[**RFC 789**] E. Rosen, “Vulnerabilities of Network Control Protocols,” RFC 789. http://www.rfc-editor.org/rfc/rfc789.txt

[**RFC 791**] J. Postel, “Internet Protocol: DARPA Internet Program Protocol Specification,” RFC 791, Sept. 1981. http://www.rfc-editor.org/rfc/rfc791.txt

[**RFC 792**] J. Postel, “Internet Control Message Protocol,” RFC 792, Sept. 1981. http://www.rfc-editor.org/rfc/rfc792.txt

[**RFC 793**] J. Postel, “Transmission Control Protocol,” RFC 793, Sept. 1981. http://www.rfc-editor.org/rfc/rfc793.txt

[**RFC 801**] J. Postel, “NCP/TCP Transition Plan,” RFC 801 Nov. 1981. http://www.rfc-editor.org/rfc/rfc801.txt

[**RFC 821**] J. Postel, “Simple Mail Transfer Protocol,” RFC 821, Aug. 1982. http://www.rfc-editor.org/rfc/rfc821.txt Obsoleted by RFC 2821.

[**RFC 822**] D. H. Crocker, “Standard for the Format of ARPA Internet Text Messages,” RFC 822, Aug. 1982. http://www.rfc-editor.org/rfc/rfc822.txt

[**RFC 826**] D. C. Plummer, “An Ethernet Address Resolution Protocol—or—Converting Network Protocol Addresses to 48.bit Ethernet Address for Transmission on Ethernet Hardware,” RFC 826, Nov. 1982. http://www.rfc-editor.org/rfc/rfc826.txt.

[**RFC 829**] V. Cerf, “Packet Satellite Technology Reference Sources,” *RFC 829*, November 1982. http://www.rfc-editor.org/rfc/rfc829.txt

[**RFC 854**] J. Postel and J. Reynolds, “TELNET Protocol Specification,” RFC 854. May 1993. http://www.rfc-editor.org/rfc/rfc854.txt

[**RFC 904**] D. Mills, “Exterior Gateway Protocol Formal Specification,” RFC 904, Apr. 1984. http://www.rfc-editor.org/rfc/rfc904.txt

[**RFC 950**] J. Mogul, J. Postel, “Internet Standard Subnetting Procedure,” RFC 950, Aug. 1985. http://www.rfc-editor.org/rfc/rfc950.txt.

[**RFC 959**] J. Postel and J. Reynolds, “File Transfer Protocol (FTP),” RFC 959, Oct. 1985. http://www.rfc-editor.org/rfc/rfc959.txt

[**RFC 977**] B. Kantor and P. Lapsley, “Network News Transfer Protocol,” RFC 977, Feb. 1986. http://www.rfc-editor.org/rfc/rfc977.txt

[**RFC 1028**] J. Davin, J.D. Case, M. Fedor, M. Schoffstall, “A Simple Gateway Monitoring Protocol,” RFC 1028, Nov. 1987, http://www.rfc-editor.org/rfc/rfc1028.txt.

[**RFC 1034**] P. V. Mockapetris, “Domain Names—Concepts and Facilities,” RFC 1034, Nov. 1987. http://www.rfc-editor.org/rfc/rfc1034.txt

[**RFC 1035**] P. Mockapetris, “Domain Names—Implementation and Specification,” RFC 1035, Nov. 1987. http://www.rfc-editor.org/rfc/rfc1035.txt

[**RFC 1058**] C. L. Hendrick, “Routing Information Protocol,” RFC 1058, June 1988. http://www.rfc-editor.org/rfc/rfc1058.txt

[**RFC 1071**] R. Braden, D. Borman, and C. Partridge, "Computing The Internet Checksum," RFC 1071, Sept. 1988. http://www.rfc-editor.org/rfc/rfc1071.txt

[**RFC 1075**] D. Waitzman, C. Partridge, S. Deering, "Distance Vector Multicast Routing Protocol," RFC 1075, Nov. 1988. http://www.rfc-editor.org/rfc/rfc1075.txt.

[**RFC 1112**] S. Deering, "Host Extension for IP Multicasting," RFC 1112, Aug. 1989. http://www.rfc-editor.org/rfc/rfc1112.txt

[**RFC 1122**] R. Braden, "Requirements for Internet Hosts—Communication Layers," RFC 1122, Oct. 1989. http://www.rfc-editor.org/rfc/rfc1122.txt

[**RFC 1123**] R. Braden, ed., "Requirements for Internet Hosts—Application and Support," *RFC-1123*, October 1989. ftp://ftp.rfc-editor.org/in-notes/rfc1123.txt.

[**RFC 1142**] D. Oran, "OSI IS-IS Intra-domain Routing Protocol," RFC 1142, Feb. 1990, ftp://ftp.rfc-editor.org/in-notes/rfc1142.txt

[**RFC 1180**] T. Socolofsky and C. Kale, "A TCP/IP Tutorial," RFC 1180, Jan. 1991. http://www.rfc-editor.org/rfc/rfc1180.txt

[**RFC 1190**] C. Topolcic, "Experimental Internet Stream Protocol: Version 2 (ST-II)," RFC 1190, October 1990. ftp://ftp.rfc-editor.org/in-notes/rfc1190.txt

[**RFC 1191**] J. Mogul, S. Deering, "Path MTU Discovery," RFC 1191, November 1990, ftp://ftp.rfc-editor.org/in-notes/rfc1191.txt

[**RFC 1213**] K. McCloghrie, M. T. Rose, "Management Information Base for Network Management of TCP/IP-based internets: MIB-II," RFC 1213, Mar. 1991. http://www.rfc-editor.org/rfc/rfc1213.txt

[**RFC 1256**] S. Deering, "ICMP Router Discovery Messages," RFC 1256, Sept. 1991. http://www.rfc-editor.org/rfc/rfc1256.txt.

[**RFC 1320**] R. Rivest, "The MD4 Message-Digest Algorithm," RFC 1320, Apr. 1992. http://www.rfc-editor.org/rfc/rfc1320.txt

[**RFC 1321**] R. Rivest, "The MD5 Message-Digest Algorithm," RFC 1321, Apr. 1992. http://www.rfc-editor.org/rfc/rfc1321.txt

[**RFC 1323**] V. Jacobson, S. Braden, and D. Borman, "TCP Extensions for High Performance," RFC 1323, May 1992. http://www.rfc-editor.org/rfc/rfc1323.txt

[**RFC 1332**] G. McGregor, "The PPP Internet Protocol Control Protocol (IPCP)," RFC 1332, May 1992. http://www.rfc-editor.org/rfc/rfc1332.txt

[**RFC 1378**] B. Parker, "The PPP AppleTalk Control Protocol (ATCP)," RFC 1378, Nov. 1992. http://www.rfc-editor.org/rfc/rfc1378.txt

[**RFC 1422**] S. Kent, "Privacy Enhancement for Internet Electronic Mail: Part II: Certificate-Based Key Management," RFC 1422, Feb. 1993. http://www.rfc-editor.org/rfc/rfc1422.txt

[**RFC 1510**] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, Sept. 1993. http://www.rfc-editor.org/rfc/rfc1510.txt

[**RFC 1519**] V. Fuller, T. Li, J. Yu, K. Varadhan, "Classless inter-domain routing (CIDR)," RFC 1519, Sept. 1993. http://www.rfc-editor.org/rfc/rfc1519.txt

[**RFC 1542**] W. Wimer, "Clarifications and Extensions for the Bootstrap Protocol," RFC 1542, Oct. 1993. http://www.rfc-editor.org/rfc/rfc1542.txt

[**RFC 1547**] D. Perkins, "Requirements for an Internet Standard Point-to-Point Protocol," RFC 1547, Dec. 1993. http://www.rfc-editor.org/rfc/rfc1547.txt

[**RFC 1584**] J. Moy, "Multicast Extensions to OSPF," RFC 1584, Mar. 1994. http://www.rfc-editor.org/rfc/rfc1584.txt

[**RFC 1631**] K. Egevang, P. Francis, "The IP Network Address Translator (NAT)," RFC 1631, May 1994. http://www.rfc-editor.org/rfc/rfc1631.txt

[**RFC 1633**] R. Braden, D. Clark, S. Shenker, "Integrated Services in the Internet Architecture: an Overview," RFC 1633, June 1994. http://www.rfc-editor.org/rfc/rfc1633.txt

[**RFC 1636**] R. Braden, D. Clark, S. Crocker, C. Huitema, "Report of IAB Workshop on Security in the Internet Architecture," RFC 1636, Nov. 1994. http://www.rfc-editor.org/rfc/rfc1636.txt

[**RFC 1661**] W. Simpson (ed.), "The Point-to-Point Protocol (PPP)," RFC 1661, July 1994. http://www.rfc-editor.org/rfc/rfc1661.txt

[**RFC 1662**] W. Simpson (ed.), "PPP in HDLC-like framing," RFC 1662, July 1994. http://www.rfc-editor.org/rfc/rfc1662.txt

[**RFC 1700**] J. Reynolds and J. Postel, "Assigned Numbers," RFC 1700, Oct. 1994. http://www.rfc-editor.org/rfc/rfc1700.txt

[**RFC 1730**] M. Crispin, "Internet Message Access Protocol—Version 4," RFC 1730, Dec. 1994. http://info.internet.isi.edu/in-notes/rfc/files/rfc1730.txt.

[**RFC 1752**] S. Bradner, A. Mankin, "The Recommendations for the IP Next Generation Protocol," RFC 1752, Jan. 1995. http://www.rfc-editor.org/rfc/rfc1752.txt

[**RFC 1760**] N. Haller, "The S/KEY One-Time Password System," RFC 1760, Feb. 1995. http://www.rfc-editor.org/rfc/rfc1760.txt

[**RFC 1762**] S. Senum, "The PPP DECnet Phase IV Control Protocol (DNCP)," RFC 1762, Mar. 1995. http://www.rfc-editor.org/rfc/rfc1762.txt.

[**RFC 1771**] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, Mar. 1995. http://www.rfc-editor.org/rfc/rfc1771.txt

[**RFC 1772**] Y. Rekhter and P. Gross, "Application of the Border Gateway Protocol in the Internet," RFC 1772, Mar. 1995. http://www.rfc-editor.org/rfc/rfc1772.txt

[**RFC 1773**] P. Traina, "Experience with the BGP-4 protocol," RFC 1773, Mar. 1995. http://www.rfc-editor.org/rfc/rfc1773.txt.

[**RFC 1779**] S. Kille, "A String Representation of Distinguished Names," RFC 1779, Mar. 1995. http://www.rfc-editor.org/rfc/rfc1779.txt .Obsoleted by RFC 2253

[**RFC 1810**] J. Touch, "Report on MD5 Performance," RFC 1810, June 1995. http://www.rfc-editor.org/rfc/rfc1810.txt

[**RFC 1812**] F. Baker, ed., "Requirements for IP Version 4 Routers*," RFC-1812*, June 1995. ftp://ftp.rfc-editor.org/in-notes/rfc1812.txt.

[**RFC 1884**] R. Hinden, S. Deering, "IP Version 6: addressing architecture," RFC 1884, Dec. 1995. http://www.rfc-editor.org/rfc/rfc1884.txt. . Obsoleted by RFC 2373

[**RFC 1906**] J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Transport Mappings for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1906, Jan. 1996. http://www.rfc-editor.org/rfc/rfc1906.txt

[**RFC 1907**] J. Case, K. McCloghrie, M. Rose, and S. Waldbusser, "Management Information Base for Version 2 of the Simple Network Management Protocol (SNMPv2)," RFC 1907, Jan. 1996. http://www.rfc-editor.org/rfc/rfc1907.txt

[**RFC 1911**] G. Vaudreuil, "Voice Profile for Internet Mail," RFC 1911, Feb. 1996. http://www.rfc-editor.org/rfc/rfc1911.txt. Obsoleted by RFC 2421.

[**RFC 1918**] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear, "Address Allocation for Private Internets," RFC 1918, February 1996. ftp://ftp.rfc-editor.org/in-notes/rfc1918.txt

[**RFC 1930**] J. Hawkinson, T. Bates, "Guidelines for Creation, Selection, and Registration of an Autonomous System (AS)," RFC 1930, March 1996. ftp://ftp.rfc-editor.org/in-notes/rfc1930.txt

[**RFC 1938**] N. Haller, C. Metz, "A One-Time Password System," RFC 1938, May 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1938.txt

[RFC 1939] J. Myers and M. Rose, "Post Office Protocol—Version 3," RFC 1939, May 1996. http://www.rfc-editor.org/rfc/rfc1939.txt

[**RFC 1945**] T. Berners-Lee, R. Fielding, H. Frystyk, "Hypertext Transfer Protocol— HTTP/1.0," RFC 1945, May 1996 http://www.rfc-editor.org/rfc/rfc1945.txt

[**RFC 1994**] W., Simpson, "PPP Challenge Handshake Authentication Protocol (CHAP)," RFC 1994, Aug. 1996, ftp://ftp.rfc-editor.org/in-notes/rfc1994.txt

[**RFC 2001**] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms," RFC 2001, Jan. 1997. http://www.rfc-editor.org/rfc/rfc2001.txt. Obsoleted by RFC 2581.

[**RFC 2003**] C. Perkins, "IP Encapsulation within IP," RFC 2003, Oct. 1996. http://www.rfc-editor.org/rfc/rfc2003.txt

[**RFC 2004**] C. Perkins, "Minimal Encapsulation within IP," RFC 2004, Oct. 1996. http://www.rfc-editor.org/rfc/rfc2004.txt.

[**RFC 2011**] K. McCloghrie, "SNMPv2 Management Information Base for the Internet Protocol using SMIv2," RFC 2011, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2011.txt

[**RFC 2012**] K. McCloghrie, "SNMPv2 Management Information Base for the Transmission Control Protocol using SMIv2," RFC 2012, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2012.txt

[**RFC 2013**] K. McCloghrie, "SNMPv2 Management Information Base for the User Data-gram Protocol using SMIv2," RFC 2013, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2013.txt

[**RFC 2018**] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, "TCP Selective Acknowledgment Options," RFC 2018, Oct. 1996. http://www.rfc-editor.org/rfc/rfc2018.txt

[**RFC 2021**] S. Waldbusser, "Remote Network Monitoring Management Information Base Version 2 using SMIv2," RFC 2021, Jan. 1997. http://www.rfc-editor.org/rfc/rfc2021.txt

[**RFC 2045**] N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies," RFC 2045, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2045.txt

[**RFC 2046**] N. Freed, N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types," RFC 2046, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2046.txt

[**RFC 2048**] N. Freed, J. Klensin, J. Postel "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures," RFC 2048, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2048.txt

[**RFC 2050**] K. Hubbard, M. Kosters, D. Conrad, D. Karrenberg, J. Postel, "Internet Registry IP Allocation Guidelines," RFC 2050, Nov. 1996. http://www.rfc-editor.org/rfc/rfc2050.txt

[**RFC 2060**] R. Crispin, "Internet Message Access Protocol—Version 4rev1," RFC 2060, Dec. 1996. http://www.rfc-editor.org/rfc/rfc2060.txt

[**RFC 2068**] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, and T. Berners-Lee, "Hypertext Transfer Protocol—HTTP/1.1," RFC 2068, Jan. 1997. http://www.rfc-editor.org/rfc/rfc2068.txt. Obsoleted by RFC 2616.

[**RFC 2104**] H. Krawczyk, M. Bellare, R. Canetti, "HMAC: Keyed-Hashing for Message Authentication," RFC 2104, Feb. 1997. http://www.rfc-editor.org/rfc/rfc2104.txt

[**RFC 2109**] D. Kristol and L. Montulli, "HTTP State Management Mechanism," RFC 2109, Feb. 1997. http://www.rfc-editor.org/rfc/rfc2109.txt

[**RFC 2131**] R. Droms, "Dynamic Host Configuration Protocol," RFC 2131, Mar. 1997. http://www.rfc-editor.org/rfc/rfc2131.txt

[**RFC 2136**] P. Vixie, S. Thomson,Y. Rekhter, and J. Bound, "Dynamic Updates in the Domain Name System," RFC 2136, Apr. 1997. http://www.rfc-editor.org/rfc/rfc2136.txt

[**RFC 2153**] W. Simpson, "PPP Vendor Extensions," RFC 2153, May 1997. http://www.rfc-editor.org/rfc/rfc2153.txt

[**RFC 2186**] K. Claffy and D. Wessels, "Internet Caching Protocol (ICP), version 2," RFC 2186, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2186.txt

[**RFC 2189**] A. Ballardie, "Core Based Trees (CBT version 2) Multicast Routing: Protocol Specification," RFC 2189, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2189.txt

[**RFC 2201**] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture," RFC 2201, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2201.txt.

[**RFC 2205**] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP)—Version 1 Functional Specification," RFC 2205, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2205.txt

[**RFC 2210**] J. Wroclawski, "The Use of RSVP with IETF Integrated Services," RFC 2210, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2210.txt

[**RFC 2211**] J. Wroclawski, "Specification of the Controlled-Load Network Element Ser-vice," RFC 2211, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2211.txt

[**RFC 2212**] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service," RFC 2212, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2212.txt

[**RFC 2215**] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements," RFC 2215, Sept. 1997. http://www.rfc-editor.org/rfc/rfc2215.txt

[**RFC 2225**] M Laubach, J. Halpern, "Classical UP and ARP over ATM," RFC 2225, April 1998. http://www.rfc-editor.org/rfc/rfc2225.txt

[**RFC 2246**] T. Dierks and C. Allen, "The TLS Protocol," RFC 2246, Jan. 1998. http://www.rfc-editor.org/rfc/rfc2246.txt

[**RFC 2253**] M. Wahl, S. Kille, T. Howes, "Lightweight Directory Access Protocol (v3)," RFC 2253, Dec. 1997. http://www.rfc-editor.org/rfc/rfc2253.txt

[**RFC 2284**] L. Blunk, J. Vollbrecht, "PPP Extensible Authentication Protocol (EAP," RFC 2284, March 1998, ftp://ftp.rfc-editor.org/in-notes/rfc2284.txt

[**RFC 2326**] H. Schulzrinne, A. Rao, R. Lanphier, "Real Time Streaming Protocol (RTSP)," RFC 2326, Apr. 1998. http://www.rfc-editor.org/rfc/rfc2326.txt

[**RFC 2328**] J. Moy, "OSPF Version 2," RFC 2328, Apr. 1998. http://www.rfc-editor.org/rfc/rfc2328.txt

[**RFC 2362**] D. Estrin, D. Farinacci, A. Helmy, D. Thaler, S. Deering, M. Handley, V. Jacobson, C. Liu, P. Sharma, L. Wei, "Protocol Independent Multicast-Sparse Mode (PIM-SM): Protocol Specification," RFC 2362, June 1998. http://www.rfc-editor.org/rfc/rfc2362.txt

[**RFC 2373**] R. Hinden, S. Deering, "IP Version 6 Addressing Architecture," RFC 2373, July 1998. http://www.rfc-editor.org/rfc/rfc2373.txt

[**RFC 2400**] J. Postel, J. Reynolds, "Internet Official Protocol Standards," RFC 2400, Sept. 1998. http://www.rfc-editor.org/rfc/rfc2400.txt. Obsoleted by RFC 2500.

[**RFC 2401**] S. Kent, R. Atkinson, "Security Architecture for the Internet Protocol," RFC 2401, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2401.txt

[**RFC 2402**] S. Kent and R. Atkinson, "IP Authentication Header," RFC 2402, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2402.txt

[**RFC 2405**] C. Madson and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm with Explicit IV," RFC 2405, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2405.txt

[**RFC 2406**] S. Kent, R. Atkinson, "IP Encapsulating Security Payload (ESP)," RFC 2406, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2406.txt

[**RFC 2407**] D. Piper, "The Internet IP Security Domain of Interpretation for ISAKMP," RFC 2407, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2407.txt

[**RFC 2408**] D. Maughan, M. Schertler, M. Schneider, J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)," RFC 2408, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2408.txt.

[**RFC 2409**] D. Harkins, D. Carrel, "The Internet Key Exchange (IKE)," RFC 2409, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2409.txt

[**RFC 2411**] R. Thayer, N. Doraswamy, R. Glenn, "IP Security Document Road Map," RFC 2411, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2411.txt

[**RFC 2420**] H. Kummert, "The PPP Triple-DES Encryption Protocol (3DESE)," RFC 2420, Sept. 1998. http://www.rfc-editor.org/rfc/rfc2420.txt

[**RFC 2421**] G. Vaudreuil, G. Parsons, "Voice Profile for Internet Mail—version 2," RFC 2421, Sept. 1998. http://www.rfc-editor.org/rfc/rfc2421.txt

[**RFC 2427**] C. Brown, A. Malis, "Multiprotocol Interconnect over Frame Relay," RFC 2427, Sept. 1998. http://www.rfc-editor.org/rfc/rfc2427.txt

[**RFC 2437**] B. Kaliski, J. Staddon, "PKCS #1: RSA Cryptography Specifications, Version 2," RFC 2437, Oct. 1998. http://www.rfc-editor.org/rfc/rfc2437.txt

[**RFC 2453**] G. Malkin, "RIP Version 2," RFC 2453, Nov. 1998. http://www.rfc-editor.org/rfc/rfc2453.txt.

[**RFC 2460**] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification," RFC 2460, Dec. 1998. http://www.rfc-editor.org/rfc/rfc2460.txt

[**RFC 2463**] A. Conta, S. Deering, "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6)," RFC 2463, Dec. 1998. http://www.rfc-editor.org/rfc/rfc2463.txt

[**RFC 2474**] K. Nicols, S. Blake, F. Baker, D. Black, "Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers," RFC 2474, Dec. 1998. http://www.rfc-editor.org/rfc/rfc2474.txt

[**RFC 2475**] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang,W. Weiss, "An Architecture for Differentiated Services," RFC 2475, Dec. 1998. http://www.rfc-editor.org/rfc/rfc2475.txt

[**RFC 2481**] K. K. Ramakrishnan and S. Floyd, "A Proposal to Add Explicit Congestion Notification (ECN) to IP," RFC 2481, Jan. 1999. http://www.rfc-editor.org/rfc/rfc2481.txt

[**RFC 2500**] J. Reynolds, R. Braden, "Internet Official Protocol Standards," RFC 2500, June 1999. http://www.rfc-editor.org/rfc/rfc2500.txt.

[**RFC 2535**] D. Eastlake, "Domain Name System Security Extensions," RFC 2535, Mar. 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2535.txt

[**RFC 2578**] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Structure of Management Information Version 2 (SMIv2)," RFC 2578, Apr. 1999. http://www.rfc-editor.org/rfc/rfc2578.txt

[**RFC 2579**] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Textual Conventions for SMIv2," RFC 2579, Apr. 1999. http://www.rfc-editor.org/rfc/rfc2579.txt

[**RFC 2580**] K. McCloghrie, D. Perkins, J. Schoenwaelder, "Conformance Statements for SMIv2," RFC 2580, Apr. 1999. http://www.rfc-editor.org/rfc/rfc2580.txt

[**RFC 2581**] M. Allman, V. Paxson,W. Stevens, " TCP Congestion Control," RFC 2581, Apr. 1999. http://www.rfc-editor.org/rfc/rfc2581.txt

[**RFC 2582**] S. Floyd, T. Henderson, "The NewReno Modification to TCP's Fast Recovery Algorithm," RFC 2582, April 1999.. ftp://ftp.isi.edu/in-notes/rfc2582.txt

[**RFC 2597**] J. Heinanen, F. Baker,W. Weiss, J. Wroclawski, "Assured Forwarding PHB Group," RFC 2597, June 1999. http://www.rfc-editor.org/rfc/rfc2597.txt.

[**RFC 2598**] V. Jacobson, K. Nichols, K. Poduri, "An Expedited Forwarding PHB," RFC 2598, June 1999. http://www.rfc-editor.org/rfc/rfc2598.txt

[**RFC 2616**] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, R. Feilding, "Hypertext Transfer Protocol—HTTP/1.1," RFC 2616, June 1999. http://www.rfc-editor.org/rfc/rfc2616.txt

[**RFC 2638**] K. Nichols, V. Jacobson, L. Zhang, "A Two-bit Differentiated Services Architecture for the Internet," RFC 2638, July 1999. http://www.rfc-editor.org/rfc/rfc2638.txt

[**RFC 2644**] D. Senie, "Changing the Default for Directed Broadcasts in Router," RFC 2644, Aug. 1999. http://www.rfc-editor.org/rfc/rfc2644.txt

[**RFC 2663**] P. Srisuresh, M. Holdrege, "IP Network Address Translator (NAT) Terminology and Considerations," RFC 2663. http://www.rfc-editor.org/rfc/rfc2663.txt

[**RFC 2715**] D. Thaler, "Interoperability Rules for Multicast Routing Protocols," RFC 2715, Oct. 1999. http://www.rfc-editor.org/rfc/rfc2715.txt

[**RFC 2716**] B. Aboba, D. Simon, "PPP EAP TLS Authentication Protocol," RFC 2716, Oct. 1999, ftp://ftp.rfc-editor.org/in-notes/rfc2716.txt

[**RFC 2733**] J. Rosenberg, H. Schulzrinne, "An RTP Payload Format for Generic Forward Error Correction," RFC 2733, Dec. 1999. http://www.rfc-editor.org/rfc2733.txt

[**RFC 2821**] J. Klensin, Eed., "Simple Mail Transfer Protocol," RFC 2821, April 2001, http://www.rfc-editor.org/rfc/rfc2821.txt

[**RFC 2827**] P. Ferguson, D. Senie, "Network Ingress Filtering: Defeating Denial of Service Attacks which Employ IP Source Address Spoofing," RFC 2827. May 2000. http://www.rfc-editor.org/rfc/rfc2827.txt

[**RFC 2893**] R. Gilligan, E. Nordmark "Transition Mechanisms for IPv6 Hosts and Routers," RFC 2893, Aug. 2000. http://www.rfc-editor.org/rfc/rfc2893.txt

[**RFC 2961**] L. Berger, D. Gan, G. Swallow, P. Pan, F. Tommasi, S. Molendini, "RSVP Refresh Overhead Reduction Extensions," RFC 2961, April 2001, ftp://ftp.rfc-editor.org/in-notes/rfc2961.txt

[**RFC 2988**] V. Paxson, M. Allman, "Computing TCP's Retransmission Timer," RFC 2988, Nov., 2000. ftp://ftp.isi.edu/in-notes/rfc2988.txt

[**RFC 3022**] P. Srisuresh, K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)," RFC 3022, Jan. 2001. http://www.rfc-editor.org/rfc/rfc3022.txt

[**RFC 3031**] E. Rosen, A. Viswanathan, R. Callon, "Multiprotocol Label Switching Architecture," RFC 3031, Jan. 2001. ftp://ftp.rfc-editor.org/in-notes/rfc3031.txt.

[**RFC 3032**] E. Rosen, D. Tappan, G. Fedorkow, Y. Rekhter, D. Farinacci, T. Li, A. Conta, "MPLS Label Stack Encoding," RFC 3032, Jan. 2001. ftp://ftp.rfc-editor.org/in-notes/rfc3032.txt

[**RFC 3052**] M. Eder, S. Nag, "Service Management Architectures Issues and Review," RFC 3052, Jan. 2001, http://www.rfc-editor.org/rfc/rfc3052.txt

[**RFC 3139**] L. Sanchez, K. McCloghrie, J. Saperia, "Requirements for Configuration Management of IP-Based Networks, RFC 3139, June 2001, http://www.rfc-editor.org/rfc/rfc3139.txt

[**RFC 3209**] D. Awduche, L. Berger, D. Gan, T. Li, V. Srinivasan, G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels," RFC 3209, Dec. 2001. ftp://ftp.rfc-editor.org/in-notes/rfc3209.txt.

[**RFC 3221**] G. Huston, "Commentary on Inter-Domain Routing in the Internet," RFC 3221, December 2001. ftp://ftp.rfc-editor.org/in-notes/rfc3221.txt

[**RFC 3232**] J. Reynolds, "Assigned Numbers: RFC 1700 is Replaced by an Online Database," RFC 3232, January 2002, http://www.rfc-editor.org/rfc/rfc3232.txt

[**RFC 3260**] D. Grossman, "New Terminology and Clarifications for Diffserv," RFC 3260, April 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3260.txt

[**RFC 3261**] J. Rosenberg, H. Schulzrinne, G. Carmarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "SIP: Session Initiation Protocol," RFC 3261, July 2002. http://www.rfc-editor.org/rfc/rfc3261.txt

[**RFC 3344**] C. Perkins, ed., "IP Mobility Support for IPv4," *RFC 3344*, October 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3344.txt

[**RFC 3346**] J. Boyle, V. Gill, A. Hannan, D. Cooper, D. Awduche, B. Christian, W. S. Lai, "Applicability Statement for Traffic Engineering with MPLS," RFC 3346, Aug. 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3346.txt

[**RFC 3376**] B. Cain, S. Deering, I. Kouvelas, B. Fenner, A. Thyagarajan, "Internet Group Management Protocol, Version 3," RFC 3376, October 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3376.txt

[**RFC 3390**] M. Allman, S. Floyd, C. Partridge, "Increasing TCP's Initial Window," RFC 3390, October 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3390.txt.

[**RFC 3410**] J. Case, R. Mundy, D. Partain, D. Partain, "Introduction and Applicability Statements for Internet Standard Management Framework," RFC 3410, December, 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3410.txt

[**RFC 3411**] D. Harrington R. Presuhn B. Wijnen, "An Architecture for Describing Simple Network Management Protocol (SNMP) Management Frameworks," RFC 3411, December 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3411.txt

[**RFC 3414**] U. Blumenthal, U. Blumenthal, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC 3414, December 2002,

[**RFC 3415**] B. Wijnen, R. Presuhn, K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)," RFC 3415, December 2002. ftp://ftp.rfc-editor.org/in-notes/rfc3415.txt

[**RFC 3416**] R. Presuhn, J. Case, K. McCloghrie, M. Rose, S. Waldbusser, "Version 2 of the Protocol Operations for the Simple Network Management Protocol (SNMP)," December 2002, ftp://ftp.rfc-editor.org/in-notes/rfc3416.txt

[**RFC 3468**] L. Andersson, G. Swallow, "The Multiprotocol Label Switching (MPLS) Working Group Decision on MPLS Signaling Protocols," RFC 3468, Feb. 2003. ftp://ftp.rfc-editor.org/in-notes/rfc3468.txt

[**RFC 3469**] V. Sharma, Ed., F. Hellstrand, Ed, "Framework for Multi-Protocol Label Switching (MPLS)-based Recovery," RFC 3469, Feb. 2003. ftp://ftp.rfc-editor.org/in-notes/rfc3469.txt

[**RFC 3550**] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: ATransport Protocol for Real-Time Applications," RFC 3550, July 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3550.txt

[**RFC 3569**] S. Bhattacharyya (ed.), "An Overview of Source-Specific Multicast (SSM)," RFC 3569, July 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3569.txt.

[**RFC 3588**] P. Calhoun, J. Loughney, E. Guttman, G. Zorn, J. Arkko, "Diameter Base Protocol," Sept. 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3588.txt

[**RFC 3600**] J. Reynolds, 6 S. Ginoza, 6, "Internet Official Protocol Standards," RFC 3600, November 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3600.txt

[**RFC 3649**] S. Floyd, "HighSpeed TCP for Large Congestion Windows," RFC 3649, December 2003, ftp://ftp.rfc-editor.org/in-notes/rfc3649.txt.

[**Rhee 1998**] I. Rhee, "Error Control Techniques for Interactive Low-bit Rate Video Transmission over the Internet," *Proceedings ACM SIGCOMM'98,* Vancouver BC, (Aug. 31–Sept. 4, 1998). http://www.acm.org/sigcomm/sigcomm98/tp/abs_24.html

[**Roberts 1967**] L. Roberts, T. Merril, “Toward a Cooperative Network of Time-Shared Computers,” *AFIPS Fall Conference,* Oct. 1966.

[**Rom 1990**] R. Rom, M. Sidi, *Multiple Access Protocols: Performance and Analysis,* Springer-Verlag, New York, 1990.

[Root Servers 2007] http://www.root-servers.org/

[**Rose 1996**] M. Rose, *The Simple Book: An Introduction to Internet Management, Revised Second Edition,* Prentice Hall, Englewood Cliffs, NJ, 1996.

[**Rosenberg 2000**] J. Rosenberg, L. Qiu, H. Schulzrinne, “Integrating Packet FEC into Adaptive Playout Buffer Algorithms on the Internet,” *IEEE INFOCOM 2000* (Tel Aviv, 2000).

[**Ross 1995**] K. W. Ross, *Multiservice Loss Models for Broadband Telecommunication - Networks,* Springer, Berlin, 1995.

[**Ross 2007**] K. W. Ross, PowerPoint slides on network security, http://cis.poly.edu/~ross/.

[**Rowston 2001**] A. Rowston, and P. Druschel, “Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems,” in *Proceedings of IFIP/ACM Middleware 2001*, 2001, Heidelberg, Germany, 2001.

[**RSA 1978**] R. L. Rivest, A. Shamir, and L. M. Adleman, “A method for obtaining digital signatures and public-key cryptosystems,” *Communications of the ACM,* Vol. 21, No. 2, pp. 120–126, Feb. 1978.

[**RSA Challenge 2002**] RSA Data Security Inc., “What is the RSA Secret Key Challenge?” http://www.rsasecurity.com/rsalabs/faq/2-4-4.html

[**RSA FAQ 2007**] RSA Inc., “RSA Laboratories' Frequently Asked Questions About Today's Cryptography, Version 4.1,” http://www.rsasecurity.com/rsalabs/faq

[**RSA Fast 2007**] RSA Laboratories, “How fast is RSA?” http://www.rsasecurity.com/rsalabs/faq/3-1-2.html

[**RSA Key 2007**] RSA Laboratories, “How large a key should be used in the RSA Crypto system?” http://www.rsasecurity.com/rsalabs/faq/3-1-5.html

[**Rubenstein 1998**] D. Rubenstein, J. Kurose, D. Towsley “Real-Time Reliable Multicast Using Proactive Forward Error Correction,” *Proceedings of NOSSDAV ‘98* (Cambridge, UK, July 1998). http://gaia.cs.umass.edu/pub/Rubenst98:proact.ps.gz

[**Rubin 2001**] A. Rubin, White-Hat Security Arsenal: Tackling the Threats, Addison-Wesley, 2001.

[**Saltzer 1984**] J. Saltzer, D. Reed, D. Clark, “End-to-End Arguments in System Design,” *ACM Transactions on Computer Systems (TOCS),* 2(4) (November 1984)..

[**Saroiu 2002**] S. Saroiu, K. Gummadi, R. Dunn, S. Gribble, H. Levy, “An Analysis of Internet Content Delivery Systems,” *Proc. Usenix OSDI 2002*, pp. 315–328. http://www.usenix.org/events/osdi02/tech/saroiu/saroiu_html/index.html

[**Savage 1999**] S. Savage, A. Collins, E. Hoffman, J. Snell, T. Anderson, “The End-to-End Effects of Internet Path Selection,” in *Proceedings of 1999 ACM SIGCOM* M, Boston, MA, September 1999

[**Savage 2000**] S. Savage, D. Wetherall, A. Karlin, T. Anderson, "Practical Network Support for IP Traceback, *Proceedings of the 2000 ACM SIGCOMM Conference*, (Stockholm, Sweden), August 2000, pp. 295–306, http://www.cs.washington.edu/homes/savage/papers/Sigcomm00.pdf

[**Saydam 1996**] T. Saydam and T. Magedanz, "From Networks and Network Management into Service and Service Management," *Journal of Networks and System Management,* Vol. 4, No. 4 (Dec. 1996), pp. 345–348.

[Schneier 1995] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C,* John Wiley and Sons, 1995.

[**Schulzrinne 1997**] H. Schulzrinne, "A Comprehensive Multimedia Control Architecture for the Internet," *NOSSDAV'97 (Network and Operating System Support for Digital Audio and Video),* St. Louis, Missouri; May 19, 1997.
http://www.cs.columbia.edu/~hgs/papers/Schu9705_Comprehensive.ps.gz

[**Schulzrinne-RTP 2007**] Henning Schulzrinne's RTP site, http://www.cs.columbia.edu/~hgs/rtp

[**Schulzrinne-RTSP 2007**] Henning Schulzrinne's RTSP site, http://www.cs.columbia.edu/~hgs/rtsp

[**Schulzrinne-SIP 2007**] Henning Schulzrinne's SIP site, http://www.cs.columbia.edu/~hgs/sip

[**Schurmann 1996**] G. Schurmann, "Multimedia Mail," *ACM Multimedia Systems,* Oct. 1996, pp. 281–295.

[**Schwartz 1977**] M. Schwartz, *Computer-Communication Network Design and Analysis*, Prentice-Hall, Englewood Cliffs, N.J., 1997.

[**Schwartz 1980**] M. Schwartz, *Information, Transmission, Modulation, and Noise,* McGraw Hill, NY, NY 1980.

[**Schwartz 1982**] M. Schwartz, "Performance Analysis of the SNA Virtual Route Pacing Control," *IEEE Transactions on Communications,* Vol. COM-30, No. 1, (Jan. 1982), pp. 172–184.

[**Schwiebert 2001**] L. Schwiebert, S. Gupta, J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors," *ACM Mobicom 2001*, 2001, pp. 151-165.
http://citeseer.ist.psu.edu/schwiebert01research.html

[**Scourias 2001**] J. Scourias, T. Farley, "Overview of the Global System for Mobile Communications: GSM." http://www.privateline.com/PCS/GSM0.html

[**Segaller 1998**] S. Segaller, *Nerds 2.0.1, A Brief History of the Internet,* TV Books, New York, 1998.

[**Semeria 1996**] C. Semeria, "Understanding IP addressing: Everything you ever wanted to know," http://www.3com.com/nsc/501302s.html

[**Shacham 1990**] N. Shacham, P. McKenney, "Packet Recovery in High-Speed Networks Using Coding and Buffer Management," *Proc. IEEE Infocom Conference* (San Francisco, 1990), pp. 124–131..

[**Sharma 1997**] Puneet Sharma, Deborah Estrin, Sally Floyd, Van Jacobson, "Scalable Timers for Soft State Protocols," *Proc. IEEE Infocom '97 Conference,* Apr. 1997 (Kobe, Japan).

[**Shipley 2001**] P. Shipley, "Open WLANS: The Early Results of War Driving,"
http://www.dis.org/filez/openlans.pdf

[**Sidor 1998**] D. Sidor, "TMN Standards: Satisfying Today's Needs While Preparing for Tomorrow," *IEEE Communications Magazine,* Vol. 36, No. 3 (Mar. 1998), pp. 54–64.

[**Singh 1999**] S. Singh, *The Code Book: The Evolution of Secrecy from Mary, Queen of Scotsto Quantum Cryptography,* Doubleday Press, 1999.

[**SIP Software 2007**] H. Schulzrinne Software Package site, http://www.cs.columbia.edu/IRT/software

[**Skype 2007**] Skype homepage, www.skype.com

[**SMIL 2007**] W3C Synchronized Multimedia homepage, http://www.w3.org/AudioVideo

[**Snoeren 2001**] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, W. T. Strayer, "Hash-Based IP Traceback," *Proceedings of the 2001 ACM Sigcomm*, http://www.acm.org/sigcomm/sigcomm2001/p1-snoeren.pdf

[**Snort 2007**] Sourcefire Inc., Snort homepage, http://www.snort.org/

[**Solari 1997**] S. J. Solari, *Digital Video and Audio Compression,* McGraw Hill, NY, NY, 1997.

[**Solensky 1996**] F. Solensky, "IPv4 Address Lifetime Expectations," in *IPng: Internet Protocol Next Generation* (S. Bradner, A. Mankin, ed), Addison-Wesley, Reading, MA, 1996.

[**Spragins 1991**] J. D. Spragins, *Telecommunications Protocols and Design,* Addison-Wesley, Reading, MA, 1991.

[**Sprint 2007**] Sprint Corp., "Dedicated Internet Access Service Level Agreements," www.sprint.com/business/resources/dedicated_internet_access.pdf

[**Spurgeon 2002**] C. Spurgeon, "Charles Spurgeon's Ethernet Web Site," http://wwwhost.ots.utexas.edu/ethernet/ethernet-home.html

[**Srinivasan 1999**] V. Srinivasan and G. Varghese, "Fast Address Lookupp Using Controlled Prefix Expansion," *ACM Transactions Computer Systems*, Vol. 17, No. 1 (Feb 1999), pp. 1–40.

[**Stallings 1993**] W. Stallings, *SNMP, SNMP v2, and CMIP The Practical Guide to Network Management Standards,* Addison-Wesley, Reading, MA, 1993.

[**Stallings 1999**] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2,* Addison-Wesley, Reading, MA, 1999.

[**Steinder 2002**] M. Steinder, A. Sethi, "Increasing robustness of fault localization through analysis of lost, spurious, and positive symptoms," in *Proc. IEEE INFOCOM*, 2002. http://www.ieee-infocom.org/2002/papers/665.pdf

[**Stevens 1990**] W. R. Stevens, *Unix Network Programming,* Prentice-Hall, Englewood Cliffs, NJ.

[**Stevens 1994**] W. R. Stevens, *TCP/IP Illustrated, Vol. 1: The Protocols,* Addison-Wesley, Reading, MA, 1994.

[**Stevens 1997**] W.R. Stevens, *Unix Network Programming, Volume 1: Networking APIs-Sockets and XTI,* 2nd edition, Prentice-Hall, Englewood Cliffs, NJ, 1997.

[**Stewart 1999**] J. Stewart, *BGP4: Interdomain Routing in the Internet,* Addison-Wesley, 1999..

[**Stoica 2001**] I. Stoica, R. Morris, D. Karger, M.F. Kaashoek, H. Balakrishnan, "Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications," In *Proceedings of ACM SIGCOMM,* 2001, (San Diego, CA), 2001. http://www.acm.org/sigcomm/sigcomm2001/p12.html

[**Stoll 1995**] C. Stoll, The Cuckoo's Egg: Tracking a Spy Through the Maze of Computer Espionage, Pocket Books, 1995.

[**Stone 1998**] J. Stone, M. Greenwald, C. Partridge, and J. Hughes, "Performance of Check-sums and CRC's Over Real Data," *IEEE/ACM Transactions on Networking,* Vol. 6, No. 5 (Oct. 1998), pp 529–543

[**Stone 2000**] J. Stone, C. Partridge, "When Reality and the Checksum Disagree," *Proceedings of ACM SIGCOMM '00,* (Stockholm, Sweden, Aug. 2000).

[**Strayer 1992**] W. T. Strayer, B. Dempsey, A. Weaver, *XTP: The Xpress Transfer Protocol,* Addison-Wesley, Reading, MA, 1992.

[**Stubblefield 2002**] A. Stubblefield, J. Ioannidis, A. Rubin, "Using the Fluhrer, Mantin, and Shamir Attack to Break WEP," *Proceedings of the 2002 Network and Distributed Systems Security Symposium* (2002), 17–22. http://www.cs.rice.edu/~astubble/wep/wep_attack.pdf

[**Subramanian 2000**] M. Subramanian, *Network Management: Principles and Practice,* Addison-Wesley, Reading, MA, 2000.

[**Subramanian 2002**] L. Subramanian, S. Agarwal, J. Rexford, R. Katz, "Characterizing the Internet Hierarchy from Multiple Vantage Points," *Proc. 2002 IEEE Infocom*.

[**Sun 2007**] Sun Microsystems, "Solistice Enterprise Manager," http://www.sun.com/software/solstice/sem/

[**Sunshine 1978**] C. Sunshine and Y. K. Dalal, "Connection Management in Transport Protocols," *Computer Networks,* North-Holland, Amsterdam, 1978.

[**T-Mobile 2004**] T-Mobile HotSpot US Location Map, http://locations.hotspot.t-mobile.com

[**Tangmunarunkit 2001**] H. Tangmunarunkit, R. Govindan, D. Estrin, S. Shenker, "The Impact of Routing Policy on Internet Paths," *Proceedings 2001 IEEE INFOCOM*, Alaska, April 2001. http://www.isi.edu/~hongsuda/publication/info2001.ps

[**TechnOnLine 2004**] TechOnLine, "Protected Wireless Networks," online webcast tutorial, http://www.techonline.com/community/tech_topic/internet/21752

[**Teleography 2002**] Teleography—a research division of Pirmetrica, "WorldCom Controls the Most Internet Bandwidth, Connections, and Revenue," http://www.telegeography.com/press/releases/2002/10-jul-2002.html.

[**Thaler 1997**] D. Thaler and C. Ravishankar, "Distributed Center-Location Algorithms," *IEEE Journal on Selected Areas in Communications,* Vol. 15, No. 3, (Apr. 1997), pp. 291–303.

[**Think 2007**] Technical History of Network Protocols, "Cyclades," http://www.cs.utexas.edu/users/chris/think/Cyclades/index.shtml

[**Thinplanet 2002**] Thinplanet homepage, http://www.thinplanet.com/

[**Thottan 1998**] M. Thottan and C. Ji, "Proactive Anomaly Detection Using Distributed Intelligent Agents," *IEEE Network Magazine,* Vol. 12, No. 5 (Sept./ Oct. 1998), pp. 21–28.

[**Tobagi 1990**] F. Tobagi, "Fast Packet Switch Architectures for Broadband Integrated Networks," *Proc. of the IEEE,* Vol. 78, No. 1 (Jan. 1990), pp. 133–167..

[**Turner 1986**] J. Turner, "New Directions in Communications (or Which Way to the Information Age?)," *Proceedings of the Zürich Seminar on Digital Communication,* (Zurich, Switzerland, Mar. 1986), pp. 25–32.

[**Turner 1988**] J. S. Turner “Design of a Broadcast packet switching network,” *IEEE Transactions on Communications,* Vol. 36, No. 6 (June 1988), pp. 734–743.

[**Utah 2004**] Utah Division of Corporations and Commercial Codes, Digital Signature Licensing Information, http://www.commerce.state.ut.us/corporat/dsmain.htm

[**Varghese 1997**] G. Varghese and A. Lauck, “Hashed and Hierarchical Timing Wheels: Efficient Data Structures for Implementing a Timer Facility,” *IEEE/ACM Transactions on Networking,* Vol. 5, No. 6, (Dec. 1997), pp. 824–834.

[**Verisign 2007**] http://www.verisign.com

[**Verizon 2007**] Verizon Communication, *Verizon Broadband Anytime.* http://www.verizon.net/wifi/

[**Verma 2001**] D.C. Verma, *Content Distribution Networks: An Engineering Approach,* John Wiley, 2001.

[**Viterbi 1995**] A. Viterbi, *CDMA: Principles of Spread Spectrum Communication,* Addison-Wesley, Reading, MA, 1995.

[**VON 2004**] Voice on the Net, http://www.von.com

[**von Lohmann 2003**] F. von Lohmann, “Peer-to-Peer File Sharing and Copyright Law: A Primer for Developers,” *2nd International Workshop on Peer-to-Peer Systems (IPTPS ‘03),* Berkeley, 2003. http://iptps03.cs.berkeley.edu/final-papers/copyright.pdf

[**Voydock 1983**] V. L. Voydock, and S.T. Kent, “Security Mechanisms in High-Level Net-work Protocols,” *ACM Computing Surveys* Vol. 15, No. 2 (June 1983), pp. 135–171.

[**W3C 1995**] The World Wide Web Consortium, “A Little History of the World Wide Web,” 1995. http://www.w3.org/History.html

[**WAP 2004**] WAP Forum, “WAP 2.0 Technical White Paper,” http://www.wapforum.org/what/whitepapers.htm.

[**Wakeman 1992**] Ian Wakeman, Jon Crowcroft, Zheng Wang, and Dejan Sirovica, “Layering Considered Harmful,” *IEEE Network,* Jan. 1992, pp. 20–24.

[**Waldvogel 1997**] M. Waldvogel et al., “Scalable High Speed IP Routing Lookup,” *Proceedings of ACM SIGCOMM ‘97* (Cannes, France, Sept. 1997). http://www.acm.org/sigs/sigcomm/sigcomm97/papers/p182.html

[**Walker 2000**] J. Walker, “IEEE P802.11 Wireless LANs, Unsafe at Any Key Size; An Analysis of the WEP Encapsulation,” Oct. 2000, http://www.drizzle.com/~aboba/IEEE/0-362.zip

[**Weatherspoon 2000**] S. Weatherspoon, “Overview of IEEE 802.11b Security,” *Intel Technology Journal,* (2nd Quarter 2000), http://developer.intel.com/technology/itj/q22000/articles/art_5.htm

[**Web ProForum 1999**] Web ProForum, “Tutorial on H.323,” 1999. http://www.webproforum.com/h323/index.html

[**Wei 2004**] W. Wei, B. Wang, J. Kurose, D. Towsley, “Detecting and Distinguishing Wired and Wireless Packet Losses in an End-End Connection,” *Technical Report*, Dept. Computer Science, University of Massachusetts, 2004.

[**Wei 2006a**] W. Wei, C. Zhang, H. Zang, J. Kurose, and D. Towsley "Inference and Evaluation of Split-Connection Approaches in Cellular Data Networks," *Proc. Active and Passive Measurement Workshop,* (Adelaide, Australlia, Mar. 2006).

[**Wei 2006b**] D. X. Wei, C. Jin, S. H. Low, S. Hegde, "FAST TCP: Motivation, Architecture, Algorithms, Performance," IEEE/ACM Transactions on Networking, Vol.14, No.6, pp.1246-1259, Dec. 2006.

[**Weinstein 2002**] S. Weinstein, "The Mobile Internet:Wireless LAN vs. 3G Cellular Mobil *e," IEEE Communications Magazine* (February 2002). pp. 26–28.

[**Weiser 1991**] M. Weiser, "The Computer for the Twenty-First Century," *Scientific American* (September 1991): 94–10. http://www.ubiq.com/hypertext/weiser/SciAmDraft3.html

[**Wessels 2001**] D. Wessels, *Web Caching*, O'Reilly, Sebastopol, CA, 2001.

[**Wimba 2004**] Wimba homepage, http://www.wimba.com

[**Woo 1994**] T. Woo, R. Bindignavle, S. Su, and S. Lam. SNP: an interface for secure network programming. In Proceedings of 1994 Summer USENIX, pages 45—58, Boston, MA, June 1994. http://www.cs.utexas.edu/users/lam/Vita/Cpapers/WBSL94.pdf

[**Wood 2007**] L. Wood, "Lloyds Satellites Constellations," http://www.ee.surrey.ac.uk/Personal/L.Wood/constellations/iridium.html

[**Xiao 2000**] X. Xiao, A. Hannan, B. Bailey, L. Ni, "Traffic Engineering with MPLS in the Internet," *IEEE Network*, March/April 2000. http://www.cse.msu.edu/~xiaoxipe/papers/mplsTE/mpls.te.pdf

[**Youtube 2007**] Youtube homepage, www.youtube.com

[**Yahoo-MIME 1999**] Yahoo MIME WWWpage, http://dir.yahoo.com/Computers_and_Internet/Multimedia/MIME/

[**Yeager 1996**] N. J. Yeager and R. E. McGrath, *Web Server Technology,* Morgan Kaufmann Publishers, San Francisco, 1996.

[**Zegura 1997**] E. Zegura, K. Calvert, M. Donahoo, "A Quantitative Comparison of Graph-based Models for Internet Topology," *IEEE/ACM Transactions on Networking,* Vol. 5, No. 6, (Dec. 1997). http://www.cc.gatech.edu/fac/Ellen.Zegura/papers/ton-model.ps.gz.

[**Zhang 1991**] L. Zhang, S. Shenker, and D. D. Clark, "Observations on the Dynamics of a Congestion Control Algorithm: The Effects of Two Way Traffic," *Proceedings of ACM SIGCOMM '91,* Zürich, 1991. http://www1.acm.org/pubs/citations/proceedings/comm/115992/p133-zhang/

[**Zhang 1993**] L. Zhang, S. Deering, D. Estrin, S. Shenker, D. Zappala, "RSVP: A New Resource Reservation Protocol," *IEEE Network Magazine,* Vol. 7, No. 9 (Sept. 1993), pp. 8–18.

[**Zhang 1998**] L. Zhang, R. Yavatkar, Fred Baker, Peter Ford, Kathleen Nichols, M. Speer, Y. Bernet, "A Framework for Use of RSVP with Diffserv Networks," <draft-ietf-diffservrsvp-01.txt>, 11/20/1998. Work in progress.

[**Zhao 2004**] B. Y. Zhao, L. Huang, J. Stribling, S. C. Rhea, A. D. Joseph, J. Kubiatowicz, "Tapestry: A Resilient Global-scale Overlay for Service Deployment," *IEEE Journal on Selected Areas in Communications,* Vol. 22, No. 1 (Jan. 2004). http://www.cs.berkeley.edu/~adj/publications/paper-files/tapestry_jsac.pdf

[**Zimmermann 1980**] H. Zimmermann, “OSI Reference Model-The ISO Model of Architecture for Open Systems Interconnection,” IEEE Transactions on Communications, Vol. 28, No. 4 (Apr. 1980), pp. 425-432.

[**Zimmermann 2007**] P. Zimmermann, “Why do you need PGP?” http://www.pgpi.org/doc/whypgp/en/.

الملحق 1 : مسرد الاختصارات

الاختصار	المصطلح/التعبير	الترجمة
3DES	Triple-DES	معيار تشفير البيانات القياسي الثلاثي
3G	Third Generation	الجيل الثالث
4G	Fourth Generation	الجيل الرابع
AAL	ATM Adaptation Layer	طبقة التكيّف بشبكة ATM، طبقة المواءمة في شبكة ATM
ABR	Available Bit Rate	معدّل البتات المتوفر، معدّل البتات المتاح
ACK	Acknowledgment	إشعار استلام، إشعار إيجابي
ACM	Association for Computing Machinery	رابطة الآلات الحاسبة (رابطة مكائن الحوسبة)
ADSL	Asymmetric Digital Subscriber Line	خط المشترك الرقمي غير المتناظر (مختلف السعة في الاتجاهين)، خط المشترِك الرقمي غير المتماثل
AES	Advanced Encryption Standard	معيار التشفير المتطور
AF	Assured Forwarding	التمرير المضمون، التمرير المؤكد
AH	Authentication Header protocol	بروتوكول ترويسة التحقق من الهوية
AIMD	Additive Increase Multiplicative Decrease	زيادة خطية ونقصان أُسّي
AP	Access Point	نقطة الوصول
API	Application Programming Interface	واجهة برمجة التطبيقات
AQM	Active Queue Management Algorithm	خوارزمية إدارة الصف النشطة (الفعالة)
ARP	Address Resolution Protocol	بروتوكول تحويل العناوين
ARQ	Automatic Repeat reQuest	إعادة الإرسال التلقائي
ASs	Autonomous Systems	النظم المستقلة ذاتياً

ATM	Asynchronous Transfer Mode	نمط النقل غير المتزامن، نمط النقل اللاتزامني
BER	Bit Error Rate	معدل الخطأ في البتات
BERs	Basic Encoding Rules	قواعد التكويد الأساسية
Botnet	roBOT NETwork	شبكة الروبوت (ويسيطر عليها الأشرار ويستخدمونها لاختراق الشبكات وإرسال رسائل الدعاية)
BS	Base Station	محطة القاعدة (نقطة الوصول اللاسلكي)
BSS	Basic Service Set	مجموعة (طاقم) الخدمة الأساسية
CA	Certification Authority	هيئة تصديق شهادات، سلطة التصديق
CA	Collision Avoidance	تجنب الاصطدام، تفادي الاصطدام، تجنب التصادمات
CAM	Content Addressable Memory	ذاكرة معنونة بمحتوياتها
CATV, CableTV	CAble TeleVision	تليفزيون الكبل
CBC	Cipher Block Chaining	تسلسل الكتل المشفرة، تسلسل كتل الشفرة
CBR	Constant Bit Rate	معدل البتات الثابت
CD	Collision Detection	كشف (اكتشاف) الاصطدام (التصادم)
CD	Compact Disk	قرص مدمج
CDMA	Code Division Multiple Access	الوصول المتعدد بتقسيم الكود، الوصول المتعدد بتقسيم الشفرات
CDN	Content Distribution Network	شبكة توزيع المحتوى
CERN	European Organization for Nuclear Research (French: Organisation Européenne pour la Recherche Nucléaire)	المنظمة الأوربية للبحوث النووية
CI	Congestion Indication	إشارة الازدحام، بيان الازدحام
CIDR	Classless InterDomain Routing	التوجيه اللانوعي بين النطاقات

CLP	Cell-Loss Priority	بت أولوية الفقد للخلية
CMIP	Common Management Information Protocol	بروتوكول معلومات الإدارة المشترك
CMISE	Common Management Information Services Element	عنصر خدمات معلومات الإدارة المشترك
COA	Care-Of-Address	عنوان العناية
Coax	Coaxial cables	الكبلات المحورية
CRC	Cyclic Redundancy Check	فحص الفائض الدوري
CSMA	Carrier Sense Multiple Access	الوصول المتعدد بالإنصات للناقل
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance	الوصول المتعدد بالإنصات للناقل مع تجنب الاصطدام
CSMA/CD	Carrier Sense Multiple Access with Collision Detection	الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام
CTS	Clear-To-Send	يمكنك الإرسال
DARPA	Defense Advanced Research Projects Agency	وكالة مشاريع البحوث المتطورة للدفاع (بإدارة الدفاع الأمريكية)
dB	Decibel	ديسيبل (وحدة قياس)
DCCP	Datagram Congestion Control Protocol	بروتوكول التحكم في الازدحام لوحدة البيانات
DDL	Data definition language	لغة لتعريف البيانات
DDoS	Distributed Denial-of-Service attack	هجوم حجب الخدمة الموزّع
DES	Data Encryption Standard	معيار تشفير البيانات القياسي
DHCP	Dynamic Host Configuration Protocol	بروتوكول تهيئة المضيف الديناميكي
DiffSev, DS	Differentiated Services	خدمات تفاضلية
DIFS	Distributed Inter-Frame Space	فترة التباعد الموزّع بين الإطارات
DMZ	DeMilitarized Zone	منطقة منزوعة السلاح، منطقة غير مؤمنة
DNS	Domain Name Service, Domain Name System	خدمة دليل أسماء النطاقات، نظام أسماء النطاقات، خدمة أسماء النطاقات

DoS	Denial-of-Service attack	هجوم حجب الخدمة
DRM	Digital Rights Management	إدارة الحقوق الرقمية
DSL	Digital Subscriber Line	خط المشترك الرقمي
EAP	Extensible Authentication Protocol	بروتوكول التحقّق القابل للامتداد
EDGE	Enhanced Data Rates for Global Evolution	تقنية معدلات البيانات المحسنّة للتطور العام
EF	Expedited Forwarding	التمرير العاجل
EFCI	Explicit Forward Congestion Indication bit	بت للبيان الصريح للازدحام الأمامي (أي في اتجاه الوجهة)
EFF	Electronic Frontier Foundation	مؤسسة الجبهة الإلكترونية
E-mail	Electronic Mail	البريد الإلكتروني
EMS	Encrypted Master Secret	سر رئيس مشفر
ER	Explicit Rate	معدل محدد (للإرسال)
ESP	Encapsulation Security Payload	بروتوكول أمن تغليف الحمل الآجر
EWMA	Exponential Weighted Moving Average	المتوسط المتحرك بأوزان أسيّة
FCFS	First-Come-First-Served	أسلوب الخدمة أولاً للواصل أولاً، أسبقية الخدمة للواصل أولاً
FDDI	Fiber Distributed Data Interface	واجهة البيانات الموزعة عبر الألياف الضوئية
FDM	Frequency-Division Multiplexing	الإرسال المتعدد بتقسيم التردد، تجميع الإشارات بتقسيم التردد
FDMA	Frequency Division Multiple Access	الوصول المتعدد بتقسيم التردد
FEC	Forward Error Correction	التصحيح الأمامي للخطأ
FHSS	Frequency-Hopping Spread Spectrum	الطيف المنتشر بتغيير التردد
FIFO	First-In-First-Out	ما يصل أولاً يُرسَل أولاً
FR	Frame relay	ترحيل الإطارات، تحويل الإطارات

FSM	Finite State Machine	آلة الأوضاع المحدودة، آلة الحالات المحدودة
FTP	File Transfer Protocol	بروتوكول نقل الملفات
GBN	Go-Back-N protocol	بروتوكول "ارجع N للوراء"، بروتوكول العودة للوراء N
GMSC	Gateway Mobile Switching Centre	مركز التحويل لبوابة خدمات قابلية الحركة
GPRS	General Packet Radio Service	تقنية خدمة رزم الراديو العامة
GSM	Global System for Mobile Communications	النظام العالمي للاتصالات النقّالة
GUI	Graphical User Interface	واجهة المستخدم الرسومية
HDLC	High-Level Data Link Control	بروتوكول التحكم عالي المستوى في وصلة البيانات، بروتوكول المستوى العالي للتحكم في وصلة ربط البيانات
HEC	Header Error Control byte	بايت التحكم في خطأ الترويسة
HFC	Hybrid Fiber-Coaxial Cable	خليط من الألياف الضوئية والكبلات المحورية، الشبكات الهجينة ذات الألياف الضوئية والكبلات المحورية، الشبكات الهجينة
HLR	Home Location Register	سجل موقع البيت
HOL	Head-of-Line (blocking)	(حجب) مقدمة الصف (الطابور)
HSDPA/HSUPA	High Speed Downlink/Uplink Packet Access	خدمة الوصول للرزم بسرعة عالية على الوصلة الصاعدة/الهابطة
HTTP	HyperText Transfer Protocol	بروتوكول نقل صفحات الويب، بروتوكول نقل النصوص التشعبية
IANA	Internet Assigned Numbers Authority	هيئة الإنترنت للأرقام المخصصة
ICANN	Internet Corporation for Assigned Names and Numbers	شركة الإنترنت للأسماء والأعداد المُخصَّصة
ICMP	Internet Control Message Protocol	بروتوكول رسائل التحكم في الإنترنت
ID	IDentifier	معرّف

IDS	Intrusion Detection System	نظام كشف الاختراق، نظام اكتشاف الاختراق
IEEE	Institute of Electrical and Electronics Engineers	معهد مهندسي الكهرباء والإلكترونيات
IETF	Internet Engineering Task Force	فريق عمل هندسة الإنترنت
IGMP	Internet Group Management Protocol	بروتوكول إدارة المجموعات للإنترنت
IM	Instant Messaging, Instant Messages	المراسلة الفورية، الرسائل الفورية
IMAP	Internet Mail Access Protocol	بروتوكول الوصول لبريد الإنترنت
IMPs	Interface Message Processors	معالجات رسائل الواجهات
IP	Internet Protocol	بروتوكول الإنترنت
IPS	Intrusion Prevention System	نظام منع الاختراق (يقوم بترشيح وحجب حركة المرور المريبة)
IPSec	Internet Protocol Security	بروتوكول IPSec (يوفر الأمن في طبقة الشبكة)
IPTV	Internet Protocol Television, Internet Television	تليفزيون الإنترنت
IPv4	Internet Protocol Version 4	الإصدار الرابع لبروتوكول الإنترنت
IPv6	Internet Protocol Version 6	الإصدار السادس لبروتوكول الإنترنت
IPX	Internetwork Packet eXchange	(بروتوكول) تبادل الرزم بين الشبكات
ISO	International Standards Organization	المنظمة الدولية للمعايير
ISPs	Internet Service Providers	موفرو خدمة الإنترنت
ITU	International Telecommunication Union	الاتحاد الدولي للاتصالات
IV	Initialization Vector	متجه التهيئة
JMF	Java Media Framework	حزمة البرمجيات للغة جافا
LAN	Local Area Network	شبكة محلية
LCD	Local Configuration Datastore	مخزن محلي لبيانات التهيئة
LCP	Link Control Protocol	بروتوكول التحكم في الوصلة

MAC	Medium Access Control	بروتوكول التحكم في الوصول للوسط
MAC	Message Authentication Code	كود التحقّق من الرسالة
MANET	Mobile Ad hoc NETwork	شبكة النمط الخاص النقّالة
MANs	Metropolitan Area Networks	شبكات المنطقة الحضرية
MCR	Minimum Cell Rate	المعدل الأدنى لإرسال الخلايا
MFA	MPLS-FR-ATM Forum	منتدى MFA
MIB	Management Information Base	قاعدة معلومات الإدارة
MIC	Message Integrity Code	كود سلامة الرسالة
MIME	Multipurpose Internet Mail Extensions	امتدادات بريد الإنترنت متعددة الأغراض
MOSPF	Multicast OSPF	بروتوكول المسار الأقصر أولاً المفتوح للإرسال المتعدد
MPLS	Multi-Protocol Label Switching	شبكات تحويل الوسْمة متعدد البروتوكول
MS	Master Secret	سر رئيس
MSC	Mobile Switching Center	مركز التحويل النقّال
MSRN	Mobile Station Roaming Number	رقم التجول للهاتف النقال
MSS	Maximum Segment Size	الحجم الأقصى للقطعة
MST	Minimum Spanning Tree	الشجرة الممتدة بأدنى كلفة
MTU	Maximum Transmission Unit	الحجم الأقصى لوحدة النقل، حجم وحدة الإرسال القصوى
NAK	Negative Acknowledgment	إشعار استلام سلبي
NAPA	Cisco's Network Application Performance Analysis	طاقم سيسكو لتحليل أداء تطبيقات الشبكات
NAT	Network Address Translation	ترجمة عناوين الشبكة
NATs	Network Address Translators	مترجمات عناوين الشبكة
NCP	Network Control Protocol	بروتوكول التحكم في الشبكة

NI	No Increase bit	بت عدم الزيادة
NIC	Network Interface Card	بطاقة واجهة الشبكة، بطاقة المواءمة للشبكة، كرت الشبكة
NOC	Network Operation Center	مركز تشغيل الشبكة
NPL	National Physical Laboratory	مختبر الفيزياء الوطني
OC	Optical Carrier	الناقل الضوئي
OSF	Open Software Foundation	مؤسسة البرامج المفتوحة
OSI	Open Systems Interconnection	نموذج ترابط الأنظمة المفتوح
OSPF	Open Shortest Path First	بروتوكول المسار الأقصر أولاً المفتوح
P2P	Pear-to-Pear distribution	توزيع نظير لنظير
P2P	Peer-to-Peer	النظائر، الأنداد، نظير لنظير، ند لند
PBX	Private Branch eXchange	سنترال فرعي خاص
PCI	Peripheral Component Interconnect	الناقل المحلي لربط المكونات الخارجية (فتحة من فتحات التوسع القياسية في الحاسب الشخصي من نوع PCI)
PCI-X	PCI eXtended	الناقل المحلي المحسّن لربط المكونات الخارجية
PCM	Pulse Code Modulation	تضمين شفرة النبضات
PCMCIA	Personal Computer Memory Card International Association	بطاقة PCMCIA (أي طبقاً لمواصفات الاتحاد الدولي لبطاقات ذاكرات الحاسبات الشخصية)
PDAs	Personal Digital Assistants	المساعدات الرقمية الشخصية
PGP	Pretty Good Privacy protocol	بروتوكول سرية جيدة جدا (يستعمل لتأمين البريد الإلكتروني)
PIN	Personal Identification Number	الرقم الشخصي السري
Pixel, PEL	Picture Element	نقطة صورة (بيكسل)
PKI	Public Key Infrastructure	بنية تحتية للتشفير بالمفاتيح العامة

PLMN	Public Land Mobile Network	شبكة البيت النقالة ذات الأرض العامّة
PMK	Pairwise Master Key	مفتاح تزاوج رئيس
POP	Point Of Presence	نقطة التواجد
POP3	Post Office Protocol Version 3	الإصدار الثالث لبروتوكول مكتب البريد
PPP	Point-to-Point Protocol	بروتوكول التوصيل من نقطة إلى نقطة
PSTN	Public Switched Telephone Network	شبكة الهاتف العمومية المحولة
PT	Payload Type field	حقل نوع الحمولة
QoS	Quality of Service	جودة الخدمة
RAM	Random Access Memory, Read And Write Memory	ذاكرة القراءة والكتابة
RED	Random-early detection algorithm	خوارزمية "الكشف المبكّر العشوائي"
RFC	Request for Comments	طلب تعليقات
RIP	Routing Information Protocol	بروتوكول معلومات التوجيه
RM	Resource Management	إدارة الموارد
RP	Rendezvous Point	نقطة الالتقاء
RPF	Reverse Path Forwarding	تمرير المسار العكسي
RR	Round Robin	التعاقب الدوراني
RRs	Resource Records	سجلات الموارد
RSA	Rivest- Shamir-Adleman algorithm	خوارزمية RSA
RSVP	Resource ReSerVation Protocol	بروتوكول حجز الموارد
RTP	Real Time Protocol	بروتوكول الوقت الحقيقي
RTS	Request-To-Send	طلب إرسال
RTSP	Real-Time Streaming Protocol	بروتوكول تشغيل المحتوى في الوقت الحقيقي

RTT	Round Trip Time	زمن رحلة الذهاب والإياب
SA	Security Association	ارتباط الأمن
SCTP	Stream Control Transmission Protocol	بروتوكول النقل بالتحكم في مسارات البيانات
SGMP	Simple Gateway Monitoring Protocol	البروتوكول البسيط لمراقبة البوابة
SHA	Secure Hash Algorithm	خوارزمية التجزئة الآمنة
SIFS	Short Inter-Frame Spacing	فترة المباعدة القصيرة بين الإطارات
SIP	Session Initiation Protocol	بروتوكول إنشاء الجلسة
SLAs	Service Level Agreements	اتفاقيات ضمان مستوى الخدمة
SMI	Structure of Management Information	هيكل معلومات الإدارة
SMTP	Simple Mail Transfer Protocol	بروتوكول نقل رسائل البريد الإلكتروني البسيط
SNMP	Simple Network Management Protocol	بروتوكول إدارة الشبكة البسيط
SNR	Signal-to-Noise Ratio	نسبة الإشارة للضوضاء
SOHO	Small Office Home Office	شبكات سوهو (المكتب الصغير والمكتب المنزلي)
SPI	Security Parameter Index	دليل متغير الأمن
SQL	Structured Query Language	لغة الاستفسارات المهيكلة (قواعد البيانات)
SR	Selective Repeat protocol	بروتوكول "الإعادة الانتقائية"
SS	Slow Start	البداية البطيئة
SSID	Service Set IDentifier	معرّف مجموعة الخدمة
SSL	Secure Socket Layer	طبقة المقابس الآمنة (توفر الأمن في طبقة النقل لتوصيلات TCP)
SSM	Source-Specific Multicast	الإرسال الجماعي المحدد بمصدر معين
SSRC	Synchronization Source Identifier	معرّف مصدر التزامن

TCP	Transmission Control Protocol	بروتوكول التحكم في الإرسال
TDM	Time-Division Multiplexing	الإرسال المتعدد بتقسيم الوقت، تجميع الإشارات بتقسيم الوقت، الإرسال المتعدد بتقسيم الزمن
TFRC	TCP Friendly Rate Control	بروتوكول التحكم في معدل البيانات المتوائم مع TCP
TK	Temporal Key	مفتاح مؤقت
TLD	Top-Level Domain	نطاق المستوى الأعلى
TLS	Transport Layer Security	أمن طبقة النقل (نسخة معدلة بعض الشيء من الإصدار الثالث لطبقة المقابس الآمنة)
TLV	Type, Length, Value	النوع، الطول، القيمة
TOS	Type of Service	نوع الخدمة
TP	Twisted-pair copper wires	أزواج أسلاك النحاس المجدولة
TTL	Time-to-Live	فترة العُمر
UBR	Unspecified Bit Rate	معدّل البتات غير المحدّد
UDP	User Datagram Protocol	بروتوكول وحدة بيانات المُستخدِم
UPnP	Universal Plug and Play Protocol	بروتوكول "وصل وشغل" العام
URL	Uniform Resource Locator	محدد الموارد الموحد (طريقة معيارية لكتابة عناوين الموارد على الويب كعنوان صفحة ويب)
UTP 5	Unshielded Twisted Pair - Category 5	أسلاك نحاس مجدولة من الفئة الخامسة
UTPs	Unshielded Twisted Pairs	أزواج الأسلاك المجدولة المكشوفة
VBR	Variable Bit Rate	معدل البتات المتغير
VC	Virtual Circuit, Virtual Channel	دائرة افتراضية، قناة افتراضية
VCI	Virtual Circuit Identifier, Virtual Channel Identifier	معرّف الدائرة الافتراضية، معرّف القناة الافتراضية

VDSL	Very-high speed Digital Subscriber Line, Very-high bit rate Digital Subscriber Line	خط المشترك الرقمي عالي السرعة
VLR	Visitor Location Register	سجل موقع الزوار
VoIP	Voice over IP	نقل الصوت على بروتوكول الإنترنت (كمكالمات الهاتف على الإنترنت)، مكالمة هاتفية عبر الإنترنت
VPN	Virtual Private Network	شبكة افتراضية خاصة
WAN	Wide-Area Network	شبكة واسعة النطاق، شبكة منطقة واسعة
WCDMA	Wideband Code Division Multiple Access	الوصول المتعدد عريض الحيز الترددي بتقسيم الكود
WEP	Wired Equivalent Privacy	الخصوصية المكافئة للشبكات السلكية
WFQ	Weighted Fair Queuing	سياسة الانتظار الموزون العادل
WiFi	WIreless FIdelity	تقنية الدقة الرقمية (تستخدم المعايير IEEE 802.11)
WiMAX	World Interoperability for Microwave Access	تقنية التوافق العالمي للوصول عبر موجات المايكرويف (معيار 802.16 IEEE)
WLANs	Wireless Local Area Networks	شبكات محلية لاسلكية
WMNs	Wireless Mesh Networks	الشبكات اللاسلكية المشبكة، الشبكات اللاسلكية المعشقة
WPAN	Wireless Personal Area Network	شبكة اللاسلكي للمنطقة الشخصية
WWW	Web, World-Wide Web	الويب، شبكة الويب العالمية، شبكة المعلومات الدولية، الشبكة العنكبوتية
WWW	World Wide Web	الويب، شبكة الويب العالمية، شبكة المعلومات الدولية، الشبكة العنكبوتية
XCP	eXplicit Congestion control Protocol	بروتوكول التحكم في الازدحام الصريح
XOR	eXclusive OR	أو- الحصرية (عملية منطقية)

الملحق 2: مسرد المصطلحات والتعبيرات

المصطلح/التعبير	الاختصار	الفصل	الترجمة
1's complement arithmetic		3,4,5	حساب مكمل الواحد
3G services		1	خدمات الجيل الثالث
3-way handshake		8	مصافحة ثلاثية الاتجاه
Access control		9	التحكم في الوصول
Access networks		1	شبكات الوصول
Access Point	AP	6,8	نقطة الوصول
Access routers		4	موجّهات الوصول
Accumulator initialization		8	تهيئة المراكم
Acknowledgment	ACK	3	إشعار استلام، إشعار إيجابي
Active Queue Management Algorithm	AQM	4	خوارزمية إدارة الصف النشطة (الفعالة)
Active scanning		6	مسح إيجابي
Ad hoc mode		6	النمط الخاص
Adapters		5	موائمات
Adaptive delta modulation		7	تضمين دلتا التكيفي
Additive Increase Multiplicative Decrease	AIMD	3	زيادة خطية ونقصان أُسّي
Address aggregation		4	تجميع العناوين، تجميع أو دمج المسارات
Address block		4	كتلة عناوين
Address Resolution Protocol	ARP	5	بروتوكول تحويل العناوين
Addressing		2,4	العنونة
Advanced Encryption Standard	AES	8	معيار التشفير المتطور
Agent advertisement		6	الإعلان عن الوكيل

Agent capabilities		9	إمكانيات الوكيل
Agent discovery		6	اكتشاف الوكيل
Agent solicitation		6	الاستفسار عن الوكيل
ALOHA protocol		5	بروتوكول ألوها
Alternating bit protocol		3	بروتوكول البت المتناوبة
Always-on		1,2	متوفرة باستمرار، تعمل دائماً
Amplification, (magnification)		2	تضخيم، تكبير، (تجسيم)
Analog form		1	صيغة تناظرية
Analog signal		1	إشارة تناظرية
Analog telephone		4	هاتف تناظري
Anchor foreign agent		6	وكيل المرساة الأجنبي
Anomaly-based systems		8	أنظمة أساسها البحث عن الشذوذ (اللاقياسي)
Anti-snubbing		2	مانع الوقف
Append-and-hash		9	التذييل والتجزئة
Application gateway		8	بوابة التطبيق
Application layer		1	طبقة التطبيقات
Application Programming Interface	API	1,2,7	واجهة برمجة التطبيقات
Association		6	ارتباط
Assured Forwarding	AF	7	التمرير المضمون، التمرير المؤكد
Asymmetric Digital Subscriber Line	ADSL	5,6	خط المشترك الرقمي غير المتناظر (مختلف السعة في الاتجاهين)، خط المشترِك الرقمي غير المتماثل
Asymptotic value		3	قيمة تقاربية
Asynchronous		9	غير متزامن، اللاتزامني
Asynchronous Transfer Mode	ATM	3,4,5,9	نمط النقل غير المتزامن، نمط النقل اللاتزامني

ATM Adaptation Layer	AAL	5	طبقة التكيّف بشبكة ATM، طبقة المواءمة في شبكة ATM
Attenuation		6	اضمحلال، وهن (الإشارة)
Audio encoder		7	مُكوّد صوتي (سمعي)
Audio player		7	مشغّل صوتي (سمعي)
Audio server		7	خادم صوتي (سمعي)
Audio/video streaming		7	تشغيل الصوت/الفيديو
Authentication		2,4,8	التحقق من الهوية، توثيق الهوية، الاستيثاق
Authentication Header protocol	AH	8	بروتوكول ترويسة التحقق من الهوية
Authentication server		6	خادم تحقّق من الهوية
Authoritative DNS servers		2	خادمات DNS المسؤولة
Automatic Repeat reQuest	ARQ	3,5,6	إعادة الإرسال التلقائي
Autonomous Systems	ASs	4	النظم المستقلة ذاتياً
Availability		9	الإتاحية، مدى توفر الخدمة بدون انقطاع
Available Bit Rate	ABR	3,4,5	معدّل البتات المتوفر، معدّل البتات المتاح
Backbone network		2	شبكة العمود الفقري
Backbone routers		4	موجّهات شبكة العمود الفقري
Backbone tier-1 router		4	موجّه الطبقة-1 لشبكة العمود الفقري
Backoff		6	تراجع
Backplane bus		4	ناقل لوحات الربط الخلفية
Bandwidth		1,2,3	الحيز الترددي
Base Station	BS	1,6	محطة القاعدة (نقطة الوصول اللاسلكي)
Base-2 arithmetic, binary arithmetic		5	الحساب الثنائي

Baseband		5	حيز التردد الأصلي
Basic Encoding Rules	BERs	9	قواعد التكويد الأساسية
Basic Service Set	BSS	6	مجموعة (طاقم) الخدمة الأساسية
Beacon frames		6	إطارات إرشاد
Best-effort service		1,4	خدمة "أفضل جهد"
Big-endian		9	ترتيبة الطرف الأكبر
Binary search		4	بحث ثنائي
BInary digiT	bit		بت، رقم ثنائي، بتة
Bit Error Rate	BER	6	معدل الخطأ في البتات
Bit errors		3	أخطاء البتات
Bit reservoir buffering		7	مستودعات التخزين المؤقت للبتات
Blanket coverage		8	تغطية شاملة
Block ciphers		8	شفرات الكتلة
Blocking		4	إيقاف، منع، حجب، توقف
Bluetooth		6	تقنية البلوتوث
Bottleneck		2	عنق الزجاجة
Bridge		9	جسر، قنطرة، معبر
Broadcast		4	الإرسال الإذاعي
Broadcast address		5	عنوان مخصص للإذاعة
Broadcast channels		5	قنوات الإذاعة
Broadcast links		5	وصلات الإذاعة
Broadcast storm		5	عاصفة البث الإذاعي
Browser (e.g. Web Browser)		2	متصفّح (كمتصفّح الويب)
Brute-force method		8	طريقة القوة العمياء
Bucket brigade attack		8	هجوم "كتيبة الدلاء"

Buffer		2,3,4,5	ذاكرة التخزين المؤقت، الذاكرة المؤقتة، مخزن مؤقت
Buffer overflow		3	فيض المخزن المؤقت
Buffered distributors		5	موزّعات بمخازن مؤقتة
Bullet-proof		8	طريقة مضمونة
Bursts		5,7	تجمّعات، دفقات (من البيانات)
Buses		5	ناقلات
Byte stuffing		5	حشو البايتات
Cable modem		5	مودم الكبل
CAble TeleVision	CATV, CableTV	1	تليفزيون (تلفاز) الكبل
Call admission		7	قبول المكالمة
Call setup protocol		7	بروتوكول إعداد المكالمة
Canonical name		2	الاسم القانوني (الرسمي)
Care-Of-Address	COA	6	عنوان العناية
Carriage return		2	رمز بداية سطر
Carrier Sense Multiple Access	CSMA	5	الوصول المتعدد بالإنصات للناقل
Carrier Sense Multiple Access with Collision Avoidance	CSMA/CA	5,6	الوصول المتعدد بالإنصات للناقل مع تجنب الاصطدام (التصادم)
Carrier Sense Multiple Access with Collision Detection	CSMA/CD	5	الوصول المتعدد بالإنصات للناقل مع اكتشاف الاصطدام (التصادم)
Cell		3	خلية
Cell-Loss Priority	CLP	5	بت أولوية الفقد للخلية
Cellular Internet access		6	الوصول الخلوي للإنترنت
Cellular networks		6	الشبكات الخلوية
Cellular telephone networks		1	شبكات الهاتف الخلوي (ومنها GSM)
Cellular telephony		6	الاتصال الهاتفي الخلوي

Central switch		5	محوّل (مفتاح) مركزي
Centralized routing algorithm		4	خوارزمية (إجراء) توجيه مركزية
Certification Authority	CA	8	هيئة تصديق الشهادات، سلطة التصديق
Chatting		2	الدردشة، المحادثة
Checksum		3,5,8	المجموع التدقيقي، أسلوب الفحص بالجمع
Chipping rate		6	معدل التقطيع
Chipping sequence		6	سلسلة التقطيع
Choke packet		3	رزمة خنق
Chosen-plaintext attack		8	الهجوم باستخدام النص الأصلي المختار
Cipher Block Chaining	CBC	8,9	تسلسل الكتل المشفرة، تسلسل كتل الشفرة
Ciphertext-only attack		8	الهجوم باستخدام النص المشفّر وحده
Circuit switching		1	تحويل الدوائر، تبديل الدوائر
Circuit-switched networks		4	شبكات تحويل الدوائر
Cisco's Network Application Performance Analysis	NAPA	9	طاقم سيسكو لتحليل أداء تطبيقات الشبكات
Class fields		7	حقول الفئات (لحركة مرور البيانات)
Classful addressing		4	عنونة نوعية
Classless InterDomain Routing	CIDR	4	التوجيه اللانوعي بين النطاقات
Clear-To-Send	CTS	6	يمكنك الإرسال
Clients		1,2	زبائن، عملاء
Coarse-grain filtering		8	ترشيح غير دقيق
Coaxial bus		5	ناقل محوري

Coaxial cables	Coax	1	الكبلات المحورية
Code Division Multiple Access	CDMA	5,6	الوصول المتعدد بتقسيم الكود ، الوصول المتعدد بتقسيم الشفرات
Codebreakers, Crackers		8	مخترقي الشفرات
Cold start, Cold boot		9	البدء البارد ، الإقلاع البارد
Collision Avoidance	CA	1,5,6	تجنب الاصطدام، تفادي الاصطدام، تجنب التصادمات
Collision Detection	CD	6	كشف (اكتشاف) الاصطدام (التصادم)
Command generator		9	مولّد أوامر
Command prompt		2	واجهة الأوامر
Command responder		9	مستجيب للأوامر
Common Management Information Protocol	CMIP	9	بروتوكول معلومات الإدارة المشترك
Common Management Information Services Element	CMISE	9	عنصر خدمات معلومات الإدارة المشترك
Compact Disk	CD	7	قرص مدمج
Company access networks, Commercial access networks		1	شبكات الوصول التجاري
Compiler		9	مترجم
Computational complexity		4	التعقيد الحسابي
Confidentiality		8	الخصوصية ، السرية
Configuration management		9	إدارة التهيئة
Configuration parameters		9	متغيرات التهيئة
Congestion		1,3,4	ازدحام
Congestion control		1,3	التحكم في الازدحام، السيطرة على الازدحام (بالحد من معدل الإرسال عندما تكون الشبكة مزدحمة)
Congestion control mechanism		2	آلية التحكم في الازدحام

Congestion Indication	CI	3	إشارة الازدحام، بيان الازدحام
Congestion notification bit		4	بت إخطار الازدحام
Congestion window		3	نافذة الازدحام
Connection identifier		6	معرّف الاتّصال، معرّف التوصيلة
Connection setup		4	إعداد توصيلة
Connectionless protocol		2	بروتوكول لاتوصيلي، بروتوكول غير توصيلي
Connectionless service		1,4,5	خدمة لاتوصيلية، خدمة غير توصيلية
Connectionless transfer protocol		3	بروتوكول نقل غير توصيلي، بروتوكول نقل لاتوصيلي
Connection-oriented service		1,4	خدمة توصيلية
Connection-oriented transfer protocol		3	بروتوكول نقل توصيلي
Constant Bit Rate	CBR	4,5	معدل البتات الثابت
Constrained optimization problem		4	مشكلة تحقيق حل أمثل ذات محدودات
Content Addressable Memory	CAM	4	ذاكرة معنونة بمحتوياتها
Content Distribution Network	CDN	7	شبكة توزيع المحتوى
Context-aware applications		6	التطبيقات المدركة للسياق
Control connection		2	توصيلة التحكم
Core functions		7	وظائف قلب الشبكة
Correspondent		6	مراسل
Count-to-infinity problem		4	مشكلة العد لما لانهاية
Crossbar switch		4	محوّل العارضة، محوّل بمسارات متعامدة
Crossbar switching fabric		4	نسيج تحويل بمسارات متعامدة
Cyclic Redundancy Check	CRC	5,6	فحص الفائض الدوري
Data			بيانات، معطيات

Data bursts		6	دفقات (تجمعات) البيانات
Data bus		5	ناقل البيانات
Data compression		7,8	ضغط البيانات
Data connection		2	توصيلة البيانات
Data decompression		8	إزالة ضغط البيانات
Data definition language	DDL	9	لغة لتعريف البيانات
Data Encryption Standard	DES	8,9	معيار تشفير البيانات القياسي
Data feeds		4	مصادر تغذية البيانات
Data integrity		2,4	سلامة البيانات
Data link layer		1,5	طبقة ربط البيانات
Data link switches		5	محوّلات طبقة ربط البيانات
Data segments		4	قطع البيانات (الرزم الناتجة من طبقة النقل)
Data traffic		2	حركة مرور البيانات
Datagram		1,3,4	وحدة بيانات
Datagram networks		4	شبكات وحدات البيانات
Decibel	dB	6	ديسيبل (وحدة قياس)
Declared traffic profile		7	نمط حركة المرور المتفق عليه
Decrypt		4	يحل الشفّرة، يزيل الشفرة
Decryption		8	إزالة التشفير، حل الشفرة
Default gateway		4	البوابة الاعتيادية
Default name server		2	خادم الاسم الاعتيادي (الافتراضي أو التلقائي)
Default router		4	الموجّه الاعتيادي (الافتراضي أو التلقائي)
Defense Advanced Research Projects Agency	DARPA	1	وكالة مشاريع البحوث المتطورة للدفاع (بإدارة الدفاع الأمريكية)
Delay		1,3	تأخير

Delay jitter		4	التفاوت الزمني للتأخير
DeMilitarized Zone	DMZ	8	منطقة منزوعة السلاح، منطقة غير مؤمَّنة
Demodulation		6	إزالة التضمين، إزالة التعديل
Demultiplexing		1,3,5	توزيع
Denial-of-Service attack	DoS	1,4,8	هجوم حجب الخدمة
Deprecated		9	مستهجن
Desktop computer		2	حاسب (حاسوب أو كمبيوتر) مكتبي
Destination host		3	مضيف الوجهة
Dial-up links		5	وصلات المودم الهاتفي
Dial-up modem		1	المودم الهاتفي
Differentiated Services	DiffSev, DS	4,5,7	خدمات تفاضلية
Digital Rights Management	DRM	7	إدارة الحقوق الرقمية
Digital signature		8	توقيع رقمي
Digital Subscriber Line	DSL	1	خط المشترك الرقمي
Digitization		7	ترقيم
Direct routing		6	التوجيه المباشر
Directional antenna		6	هوائي اتجاهي
Directives		7	توجيهات
Dispatch module		9	وحدة إرسال
Distance-vector routing		4	توجيه متجه المسافة
Distributed Denial-of-Service attack	DDoS	1,2,3	هجوم حجب الخدمة الموزّع
Distributed games		1	الألعاب الموزّعة
Distributed Inter-Frame Space	DIFS	6	فترة التباعد الموزّع بين الإطارات
Distribution time		2	زمن التوزيع
DNS lookup		2	عملية البحث في دليل DNS

Document		2	وثيقة، مستند
Domain Name Service, Domain Name System	DNS	2,3,5,8,9	خدمة دليل أسماء النطاقات، نظام أسماء النطاقات، خدمة أسماء النطاقات
Downlink channel		5	قناة الوصلة الهابطة
Download and delete		2	يُنزّل ويحذف، جلب وحذف
Download and keep		2	يُنزّل ويحتفظ، جلب واحتفاظ
Downstream		1	الاتجاه النازل من الإنترنت
Dynamic Host Configuration Protocol	DHCP	4	بروتوكول تهيئة المضيف الديناميكي
Eavesdroppers		8	المتلصصين
Edge functions		7	وظائف حافة الشبكة
Egress point		4	نقطة خروج
Elastic applications		2	تطبيقات مرنة
Electronic Frontier Foundation	EFF	8	مؤسسة الجبهة الإلكترونية
Electronic Mail	E-mail	2	البريد الإلكتروني
Encapsulation		1	تغليف البيانات (في الطبقات المختلفة)
Encapsulation and decapsulation		6	التغليف وفك التغليف
Encapsulation Security Payload	ESP	8	بروتوكول أمن تغليف الحمل الآجر
Encoding		1	تكويد
Encrypt		4	يشفّر (البيانات)
Encrypted Master Secret	EMS	8	سر رئيس مشفر
Encryption		2,8	تشفير
Encryption keys		8	مفاتيح التشفير
End game mode		2	نمط نهاية اللعبة
End point authentication		1,8	التحقّق من هوية النقطة الطرفية

End systems		1,2,3	أنظمة طرفية
Enhanced Data Rates for Global Evolution	EDGE	6	تقنية معدلات البيانات المحسنّة للتطور العام
Entity body		2	محتوى الكيان
Error burst		5	تجمع أخطاء
Error Correction		5	تصحيح الأخطاء
Error detection		3,5	اكتشاف الأخطاء
Ethernet efficiency		5	كفاءة الإيثرنت
Ethernet network		5	شبكة الإيثرنت
European Organization for Nuclear Research (French: Organisation Européenne pour la Recherche Nucléaire)	CERN	1	المنظمة الأوربية للبحوث النووية
Even parity		5	تكافؤ زوجي
Event-driven programming		3	البرمجة المبنية على الحدث
eXclusive OR	XOR	5,7	أو- الحصرية (عملية منطقية)
Exhaustive search		8	عمليات البحث الاستقصائي
Expedited Forwarding	EF	7	التمرير العاجل
Explicit Forward Congestion Indication bit	EFCI	3	بت للبيان الصريح للازدحام الأمامي (أي في اتجاه الوجهة)
Explicit Rate	ER	3	معدل محدد (للإرسال)
Exponential backoff		5	تراجع أسّي
Exponential Weighted Moving Average	EWMA	3	المتوسط المتحرك بأوزان أسّية
Extended FSM		3	آلة الأوضاع المحدودة الموسّعة
Extensible Authentication Protocol	EAP	8	بروتوكول التحقّق القابل للامتداد
Factorization		8	تحليل للعوامل (كتحليل العدد لعوامله الأولية)
Fading		6	خفوت قوة الإشارة
Fast forward		7	التقدم بسرعة

Fast recovery		3	التعافي السريع (من أثر الازدحام)
Feedback		3,4	التغذية المرتدة، التغذية الخلفية، التغذية المرتجعة
Fiber Distributed Data Interface	FDDI	5	واجهة البيانات الموزعة عبر الألياف الضوئية
File downloading		2	تنزيل الملف، جلب الملف
File transfer		2	نقل الملفات
File Transfer Protocol	FTP	2,3,7	بروتوكول نقل الملفات
File uploading		2	تحميل الملف، إيداع الملف
Filtering		5	الترشيح
Finite State Machine	FSM	3	آلة الأوضاع المحدودة، آلة الحالات المحدودة
Firewalls		1,2,3,4,7,8,9	برامج الحماية، الجدران النارية، الحواجز المانعة
First hop router		4	موجّه أول قفزة
First-Come-First-Served	FCFS	1,7	أسلوب الخدمة أولاً للواصل أولاً، أسبقية الخدمة للواصل أولاً
First-In-First-Out	FIFO	7	ما يصل أولاً يُرسَل أولاً
Flags		2,3	أعلام، مؤشرات
Flooding		4	فيض، فيضان
Flow control		3,5	ضبط التدفق، التحكم في التدفق
Flow control window		4	نافذة التحكم في التدفق
Flow label		4	وسمة التدفق
Foreign agent		6	الوكيل الأجنبي
Foreign network		6	الشبكة الأجنبية (أو المزورة)
Forward Error Correction	FEC	5,6,7	التصحيح الأمامي للخطأ
Forwarding		4,5,7	تمرير
Forwarding table		4	جدول تمرير

Fourth Generation	4G	6	الجيل الرابع
Fragmentation		4	تجزئة
Frame relay	FR	4,5,9	ترحيل الإطارات، تحويل الإطارات
Frames		1,4	إطارات (فترات زمنية، أو رزم البيانات في طبقة ربط البيانات)
Framing		5	تأطير
Free-riding problem		2	مشكلة "الركوب المجاني"
Frequency Division Multiple Access	FDMA	6	الوصول المتعدد بتقسيم التردد
Frequency-Division Multiplexing	FDM	1,5,6	الإرسال المتعدد بتقسيم التردد، تجميع الإشارات بتقسيم التردد
Frequency-Hopping Spread Spectrum	FHSS	6	الطيف المنتشر بتغيير التردد
Full-duplex		3,5	كامل الازدواج في الاتجاهين، اتصال مزدوج الإرسال كامل
Functional components		7	المكوّنات الوظيفية
Gatekeeper		7	حارس البوابة
Gateway Mobile Switching Centre	GMSC	6	مركز التحويل لبوابة خدمات قابلية الحركة
Gateway routers		4	موجّهات البوّابة
General Packet Radio Service	GPRS	6	تقنية خدمة رزم الراديو العامة
Generator		5	مولّد
Geostationary satellites		1	أقمار ثابت بالنسبة للكرة الأرضية
Global System for Mobile Communications	GSM	1,6	النظام العالمي للاتصالات النقّالة
Go-Back-N protocol	GBN	3	بروتوكول "ارجع N للوراء"، بروتوكول العودة للوراء N
Granularity		9	درجة التجزئة
Graph		4	رسم بياني، شكل بياني

Graph theory		2	نظرية الأشكال البيانية
Graphical User Interface	GUI	1	واجهة المستخدم الرسومية
Grid computing		3	الحوسبة الشبكية
Group-shared tree		4	شجرة مشتركة للمجموعة (بغض النظر عن عدد المصادر)
Guaranteed QoS		7	جودة مضمونة للخدمة
Guided media		1	أوساط موجّهة
Half-duplex		5	اتصال مزدوج الإرسال نصفي
Handoff		6	انتقال المستخدم من نقطة اتصال بالشبكة إلى نقطة أخرى ، تغيير نقطة الارتباط بالشبكة، تغير نقطة الخدمة، تحوّل الاتصال بين نقاط الوصول، عملية مناولة
Handshaking		2,3,4,5	مصافحة
Hard guarantee		7	ضمان مؤكد
Hard state		4, 7	حالة محددة، حالة صلبة
Hardware components		1,5	المكونات المادية
Hash		8	هاش
Hash functions		8,9	دوال التجزئة، دوال التحوير
Hashed message authentication codes		9	شفرات مجزأة لتوثيق الرسائل
Header		1,3,5,8	ترويسة
Header bytes		1	بايتات الترويسة
Header checksum		4	المجموع التدقيقي للترويسة
Header Error Control byte	HEC	5	بايت التحكم في خطأ الترويسة
Header lines		2	سطور الترويسة
Head-of-Line (blocking)	HOL	4	(حجب) مقدمة الصف (الطابور)
Hidden terminal problem		6	مشكلة "المحطة الطرفية المخفية"
Hierarchical routing		4	توجيه هرمي

Hierarchical structure		2,4,5	تركيب هرمي (شجري)
High Speed Downlink/Uplink Packet Access	HSDPA/HSUPA	6	خدمة الوصول للرزم بسرعة عالية على الوصلة الصاعدة/الهابطة
High-Level Data Link Control	HDLC	5	بروتوكول التحكم عالي المستوى في وصلة البيانات، بروتوكول المستوى العالي للتحكم في وصلة ربط البيانات
High-speed LAN		2	شبكة اتصالات محلية بسرعة عالية
Hit rate		2	معدل إصابة الهدف
Home agent		6	وكيل البيت
Home Location Register	HLR	6	سجل موقع البيت
Home network		6	شبكة البيت
Hop limit		4	حد عدد القفزات
Hops		2	قفزات
Host		1,3	مضيف
Host aliases		2	أسماء المضيف البديلة
Hostname		2	اسم المضيف (مثل (kfupm.edu.sa
Hot-potato routing		4	توجيه البطاطس الساخنة
HTTP caching		5	تخزين نسخة من ملفات HTTP في الذاكرة المخبأة
Hub		5,9	مُجمّع
Hybrid Fiber-Coaxial Cable	HFC	1,5	خليط من الألياف الضوئية والكبلات المحورية، الشبكات الهجينة ذات الألياف الضوئية والكبلات المحورية، الشبكات الهجينة
Hyperlink		2,7	رابط تشعبي، وصلات تشعبية (على صفحات الويب)

Hypertext		1	النص التشعبي
HyperText Transfer Protocol	HTTP	1,2,7	بروتوكول نقل صفحات الويب، بروتوكول نقل النصوص التشعبية
IDentifier	ID	2,3,5	معرّف
Import policy		4	سياسة استيراد
In-band		2	داخل النطاق
Indexing		7	الفهرسة
Indirect routing		6	التوجيه غير المباشر
Information request message		9	رسالة طلب معلومات
Infrastructure mode		6	نمط البنية التحتية
Ingress point		4	نقطة دخول
Initialization Vector	IV	8	متجه التهيئة
Input link		4	وصلة المدخل
Input link interface		4	واجهة وصلة المدخل
Instant Messaging, Instant Messages	IM	2	المراسلة الفورية، الرسائل الفورية
Integer		9	عدد صحيح
Inter-autonomous systems, inter-AS routing		4	التوجيه بين النظم المستقلة ذاتياً
Interconnection network		4	شبكة ربط بينية
Interface		2,4	واجهة
Interface Message Processors	IMPs	1	معالجات رسائل الواجهات
Interference		1,6	التداخل (بسبب إشارات المصادر الأخرى وغيرها من الإشارات الكهرومغناطيسية)
Interior gateway protocols		4	بروتوكولات البوّابة الداخلية
International Standards Organization	ISO	1,9	المنظمة الدولية للمعايير
International Telecommunication Union	ITU	5,7,8,9	الاتحاد الدولي للاتصالات

Internet		- -	الإنترنت، الشابكة، الشبكة الدولية
Internet Assigned Numbers Authority	IANA	9	هيئة الإنترنت للأرقام المخصصة
Internet Control Message Protocol	ICMP	4,9	بروتوكول رسائل التحكم في الإنترنت
Internet Corporation for Assigned Names and Numbers	ICANN	2	شركة الإنترنت للأسماء والأعداد المُخصَّصة
Internet Engineering Task Force	IETF	3,4,5,8,9	فريق عمل هندسة الإنترنت
Internet Group Management Protocol	IGMP	4	بروتوكول إدارة المجموعات للإنترنت
Internet Mail Access Protocol	IMAP	2	بروتوكول الوصول لبريد الإنترنت
Internet Protocol	IP	1,4	بروتوكول الإنترنت
Internet Protocol Security	IPSec	8	بروتوكول IPSec (يوفر الأمن في طبقة الشبكة)
Internet Protocol Television, Internet Television	IPTV	2,7	تليفزيون الإنترنت
Internet Protocol Version 4	IPv4	4	الإصدار الرابع لبروتوكول الإنترنت
Internet Protocol Version 6	IPv6	4	الإصدار السادس لبروتوكول الإنترنت
Internet radio		7	راديو الإنترنت
Internet Service Providers	ISPs	1	موفرو خدمة الإنترنت
Internet Systems Consortium		4	اتحاد نظم الإنترنت
Internet telephony		7	هاتف الإنترنت
Internetworking		1,3,6	تربيط الشبكات، التوصيل ما بين الشبكات، التشبيك البيني
Interpolation algorithms		7	خوارزميات الاستكمال
Interprocess communication		2	الاتصال بين العمليات

Interrupt signal		3,4,5	إشارة مقاطعة
Intra-autonomous systems, Intra-AS routing		4	التوجيه داخل النظم المستقلة ذاتياً
Intra-domain routing protocol		9	بروتوكول التوجيه داخل النطاق
Intranet		1	الإنترانت
Intrusion Detection System	IDS	4,8	نظام كشف الاختراق، نظام اكتشاف الاختراق
Intrusion Prevention System	IPS	8	نظام منع الاختراق (يقوم بترشيح وحجب حركة المرور المريبة)
IP addresses		5	عناوين طبقة الشبكة
IP multicast transmission		7	إرسال IP المتعدد
IP spoofing		1,8	تزييف عنوان IP، انتحال عنوان IP
Iterative query		2	الاستفسار التكراري (كما في DNS)
Java Media Framework	JMF	7	حزمة البرمجيات للغة جافا
Keywords		2	الكلمات الدليلية
Known-plaintext attack		8	الهجوم باستخدام النص الأصلي المعروف
Label		5	وسْمة
Label-switched router		5	موجّه التحويل بوسْمة
LAN address		5	عنوان الشبكة المحلية
LAN-on-motherboard		5	شبكة محلية على اللوحة الأم
Laptops		1,2,6	حاسبات نقالة، حاسبات محمولة
Large-scale file sharing		2	مشاركة الملفات على نطاق واسع
Latency		9	مدى التأخير في الإرسال والاستقبال
Layered architecture		1	البنية المعمارية الطبقية
Leaky bucket		7	آلية الدلو المثقوب

Least significant byte first		9	أسلوب التخزين بالبايت الأدنى أولاً
Line card		4,5	بطاقة الخط، كرت الخط، بطاقة واجهة الشبكة
Line feed		2	تغذية سطر جديد
Linear broadcast bus		5	ناقل إذاعة خطي
Linear predictive encoding		7	التضمين التنبؤي الخطيّ
Link Control Protocol	LCP	5	بروتوكول التحكم في الوصلة
Link utilization		9	مدى استغلال الوصلة
Link-layer switches		1	محوّلات طبقة ربط البيانات
Link-level scheduling		7	جدولة على مستوى الوصلة
Links		1,5	وصلات
Link-state routing		4	توجيه حالة الوصلة
Little-endian		9	ترتيبة الطرف الأصغر
Load balancing		2	توازن الأحمال
Local Area Network	LAN	1,5	شبكة محلية
Local Configuration Datastore	LCD	9	مخزن محلي لبيانات التهيئة
Location-aware applications		6	التطبيقات المدركة للمكان
Log file		8	ملف النشاطات
Logical communication		3	اتصالاً منطقياً
Longest prefix matching rule		4	قاعدة تطابق البادئة الأطول
Long-lived routing loop		4	حلقة توجيه طويلة الأمد
Lookup		4	عملية البحث في الجدول
Loss		1	الفقد
Loss anticipation schemes		7	أساليب توقع الفقد
Low-earth orbiting (LEO) satellites		1	أقمار تدور حول الأرض في مدار منخفض
Luminance		7	الإضاءة

MAC address		5	عنوان طبقة ربط البيانات (عنوان الماك)
Mail folders		2	مجلدات البريد
Mail server aliasing		2	أسماء خادم البريد البديلة
Malware		1,4	برمجيات خبيثة، برامج خبيثة
Managed objects		9	كائنات مُدارة
Management Information Base	MIB	9	قاعدة معلومات الإدارة
Manchester coding		5	تكويد مانشستر
Man-in-the-middle attack		1,2,8	هجوم "رَجُل في الوسط"
Many-to-many		7	من العديد إلى العديد
Marking		7	التعليم
Master node		5,6	عقدة رئيسة (السيد)
Master Secret	MS	8	سر رئيس
Maximum Segment Size	MSS	3	الحجم الأقصى للقطعة
Maximum Transmission Unit	MTU	4,5	الحجم الأقصى لوحدة النقل، حجم وحدة الإرسال القصوى
Media player		7	مشغّل مواد وسائط متعددة
Medium Access Control	MAC	5	بروتوكول التحكم في الوصول للوسط
Membership		4	عضوية
Memory access		4	الوصول للذاكرة
Message authentication		8	التحقّق من الرسالة، توثيق الرسالة
Message Authentication Code	MAC	8	كود التحقّق من الرسالة
Message digest		8,9	مختصر بسيط للرسالة
Message ID		9	معرِّف الرسالة
Message integrity		8	سلامة الرسالة
Message Integrity Code	MIC	9	كود سلامة الرسالة

Metafile		7	ملف تعريفي
Metering		7	وظيفة القياس
Metropolitan Area Networks	MANs	5	شبكات المنطقة الحضرية
MIB modules		9	وحدات MIB
Minimum Cell Rate	MCR	4	المعدل الأدنى لإرسال الخلايا
Minimum Spanning Tree	MST	4	الشجرة الممتدة بأدنى كلفة
Minimum threshold		4	عتبة الحد الأدنى
Mobile Ad hoc NETwork	MANET	6	شبكة النمط الخاص النقّالة
Mobile IP		6	بروتوكول الإنترنت النقّال
Mobile networks		6	الشبكات النقّالة، شبكات المحمول
Mobile phone		2	هاتف جوال، محمول، جوال
Mobile Station Roaming Number	MSRN	6	رقم التجول للهاتف (للمحطة) النقال
Mobile Switching Center	MSC	6	مركز التحويل (للهاتف) النقّال
Mobility		1,6	قابلية الحركة
Mobility-assisting agents		1	وكلاء المساعدة لقابلية الحركة
Modem		5	مودم
Modulation		1,6	تضمين، تعديل
Module compliance		9	توافق الوحدة
Module identity		9	هوية الوحدة
Modulo-2 arithmetic		3,5	حساب الباقي الثنائي
Mono		7	أحادي
Monoalphabetic encryption		8	تشفير أحادي الحرف
Most significant byte first		9	أسلوب التخزين بالبايت الأعلى أولاً
Motherboard		5	اللوحة الأم
MPLS-FR-ATM Forum	MFA	5	منتدى MFA

Multicast		4	الإرسال الجماعي، الإرسال المتعدد
Multicast OSPF	MOSPF	4	بروتوكول المسار الأقصر أولاً المفتوح للإرسال المتعدد
Multicast Overlay Networks		7	شبكات إرسال متعدد إضافية
Multicast service		4	خدمة الإرسال المتعدد (الجماعي)
Multicast trees		7	شجرات الإرسال المتعدد
Multimedia		2	الوسائط المتعددة
Multimedia applications		2	تطبيقات الوسائط المتعددة (كعرض شرائط الفيديو وراديو الإنترنت وهاتف الإنترنت)
Multimode fiber optic cables		1	كبلات الألياف الضوئية متعدّدة الأنماط
Multi-path fading		1	ضعف الإشارة بسبب وصولها عبر مسارات متعدّدة (نتيجة انعكاس الموجات على الأجسام المختلفة)
Multipath propagation		6	الانتقال متعدد المسار
Multiple access		5	الوصول المتعدد
Multiplexing		1,3,5	تجميع
Multi-Protocol Label Switching	MPLS	5	شبكات تحويل الوسْمة متعدد البروتوكول
Multipurpose Internet Mail Extensions	MIME	2	امتدادات بريد الإنترنت متعددة الأغراض
National Physical Laboratory	NPL	1	مختبر الفيزياء الوطني
Negative Acknowledgment	NAK	3	إشعار استلام سلبي
Network adapter card		5	بطاقات مواءمة الشبكة
Network Address Translation	NAT	4	ترجمة عناوين الشبكة
Network Address Translators	NATs	2	مترجمات عناوين الشبكة
Network Associates		8	اتحاد الشبكات

Network Control Protocol	NCP	1	بروتوكول التحكم في الشبكة
Network core devices		2	أجهزة قلب الشبكة (كالموجّهات والمحوّلات)
Network dimensioning		7	تخطيط الشبكة
Network flow problem		4	مشكلة تدفق الشبكة
Network infrastructure		2	البنية التحتية للشبكة
Network Interface Card	NIC	4,5	بطاقة واجهة الشبكة، بطاقة المواءمة للشبكة، كرت الشبكة
Network layer		1	طبقة الشبكة
Network management agent		9	وكيل إدارة شبكة
Network Operation Center	NOC	9	مركز تشغيل الشبكة
Network security		1,8	أمن الشبكات، تأمين الشبكة
Network sniffing		8	إلتقاط الرزم من الشبكة
Network topology		1	طبوغرافية الشبكة، طبولوجيا الشبكة
Newsgroups		2	مجموعات الأخبار
No Increase bit	NI	3	بت عدم الزيادة
Nonce		8,9	العدد الذي يستخدمه البروتوكول مرّة واحدة فقط
Non-persistent connections		2	التوصيلات غير الدائمة
Notification		9	إخطار
Notification originator		9	مُنشئ إخطارات، مولّد إخطارات
Notification receiver		9	مُستقبل إخطارات
Object		2	كائن، شيء
Object identifier		9	مُعرِّف كائن
Object type		9	نوع الكائن
Obsolete		9	متقادم

Octet string		9	سلسلة بايتات
Odd parity		5	تكافؤ فردي
Offered load		3	الحمل المقدم (للشبكة)
Omnidirectional antenna		6	هوائي شامل لكل الاتجاهات
One-to-many		7	من واحد إلى العديد
One-to-one mapping		8	تناظر واحد لواحد
Open Shortest Path First	OSPF	4	بروتوكول المسار الأقصر أولاً المفتوح
Open Software Foundation	OSF	9	مؤسسة البرامج المفتوحة
Open source code		4	برنامج مصدر مفتوح
Open Systems Interconnection	OSI	1	نموذج ترابط الأنظمة المفتوح
Opportunistic scheduling		6	الجدولة الانتهازية
Optical Carrier	OC	1	الناقل الضوئي
Optimal route		4	المسار الأمثل
Origin authentication		4	توثيق المصدر، التحقق من هوية المصدر
OS kernel		8	لبّ نظام التشغيل
Outage notification		9	الإخطار بانقطاع الخدمة
Out-of-band		7,8	خارج النطاق
Output link		4	وصلة مخرج
Output link interface		4	واجهة وصلة مخرج
Overflow		3	فيض
Overhead		4	عبء إضافي
Overlay network		2,4,5,7	شبكة إضافية، شبكة الغطاء
P2P file sharing		1,2	مشاركة النظائر للملفات
Packet		1	رزمة، حزمة
Packet classification		7	تصنيف الرزم

Packet delay		7	تأخير الرزم
Packet filters		2	مرشحات الرزم
Packet fragmentation		4	تجزئة الرزم
Packet jitter		7	التذبذب في تأخير الرزم، التفاوت في تأخير الرزم
Packet radio networks		1	شبكات رزم الراديو
Packet satellite		1	الرزم للأقمار الصناعية
Packet scheduler		4	مُجدول الرزم
Packet signature		4	توقيع الرزمة، بصمة الرزمة
Packet sniffing		1	التقاط الرزم
Packet switch		1,4	محوّل رزم
Packet switching		1	تحويل رزم البيانات
Packetization		7	الترزيم
Packet-switched networks		4	شبكات تحويل الرزم
Padding		8	حشو الأصفار
Pairwise Master Key	PMK	8	مفتاح تزاوج رئيس
Palmtops		6	حاسبات الكف
Parity check		5	أسلوب فحص التكافؤ (لاكتشاف الأخطاء)
Parked device		6	جهاز مستوقف
Passive scanning		6	مسح سلبي
Password		8	كلمة سر، كلمة مرور
Patch		2	ترميم، رقعة
Path loss		6	خسارة المسار، الفقد
Pause		7	توقف مؤقت
Payload		1,3,6,8,9	الحمل الآجر
Payload Type field	PT	5	حقل نوع الحمولة

PCI eXtended	PCI-X	5	الناقل المحلي المحسّن لربط المكونات الخارجية
Peak rate		7	معدّل إرسال الذروة
Peer-to-Peer	P2P	2	النظائر، الأنداد، نظير لنظير، ند لند
Pear-to-Pear distribution	P2P	7	توزيع نظير لنظير
Per-element call admission		7	قبول المكالمات على مستوى العنصر
Per-hop behavior		7	السلوك على مستوى القفزة
Periodicity		4	الدورية، التكرارية
Peripheral Component Interconnect	PCI	5	الناقل المحلي لربط المكونات الخارجية (فتحة من فتحات التوسع القياسية في الحاسب الشخصي من نوع PCI)
Persistent connections		2	التوصيلات الدائمة
Personal Computer Memory Card International Association	PCMCIA	5	بطاقة PCMCIA (أي طبقاً لمواصفات الاتحاد الدولي لبطاقات ذاكرات الحاسبات الشخصية)
Personal Digital Assistants	PDAs	1,2,6	المساعدات الرقمية الشخصية
Personal Identification Number	PIN	8	الرقم الشخصي السري
Phoneme		7	فونيم صوتي
Physical address		5	العنوان المادي
Physical layer		1	الطبقة المادية
Piconet		6	شبكة مصغرة
Picture Element	Pixel, PEL	7	نقطة صورة (بيكسل)
Piggybacked		3	"راكب على ظهر" (تعبير يستخدم عند إرسال إشعار الاستلام مع البيانات في نفس الرزمة)

Pipeline		3	خط الأنابيب
Pipelining		2	طريقة خط الأنابيب (للتشغيل المتوازي)
Plaintext, cleartext		4,8	النص الأصلي
Platform		8	منصة
Playback attacks		9	هجمات إعادة التشغيل
Playout delay		7	تأخير الاستماع
Plug-and-play		5	وصّل وشغّل
Point Of Presence	POP	1,9	نقطة التواجد
Point-to-point link		5,6	وصلة اتصال من نقطة إلى نقطة
Point-to-Point Protocol	PPP	1,5	بروتوكول التوصيل من نقطة إلى نقطة
Poisoned Reverse		4	اتجاه عكسي مسمّم
Polling		6,9	استفتاء، استطلاع
Polling protocol		5	بروتوكول الاستفتاء
Polyalphabetic encryption		8	تشفير متعدد الحروف
Polynomials		5	دوال متعددة الحدود، كثيرات الحدود
Port scanners		3	ماسحات المنافذ
Port scanning		8	مسح المنافذ
Post Office Protocol Version 3	POP3	2	الإصدار الثالث لبروتوكول مكتب البريد
Preamble		5	بتات الديباجة (الاستهلال)
Predecessor		4	سلف، سابق
Prefetch		7	الجلب المبكّر
Prefix		4,7	بادئة، سابقة
Presentation layer		9	طبقة التقديم
Presentation service		9	خدمة تقديم

Pretty Good Privacy protocol	PGP	8	بروتوكول سرية جيدة جدا (يستعمل لتأمين البريد الإلكتروني)
Prime numbers		8	أعداد أولية
Privacy		2	الخصوصية، السرية
Private Branch eXchange	PBX	7	سنترال فرعي خاص
Proactive anomaly detection		9	الكشف الاستباقي للأداء الشاذ (اللاقياسي)
Probing		6	تقصّي
Probing peer		2	نظير المجس، نظير التقصّي
Processes		2,3	عمليات
Processing delay		2	زمن المعالجة
Propagation delay, Propagation time		2	زمن الانتقال
Proprietary		2	ملكية خاصة
Proprietary networks		1	الشبكات ذات الملكية الخاصة
Proprietary protocols		7	بروتوكولات مملوكة لشركات، بروتوكولات ذات ملكية خاصة
Protocols		--	بروتوكولات، مراسم
Protocol stack		1,2,5	رصة البروتوكولات
Proxy		2	وكيل، مفوض، بروكسي، مساعد
Proxy forwarder		9	ممرِّر بالوكالة
Proxy server		2	الخادم المفوّض (الوكيل)
Pruning		4	تقليم
Psychoacoustic masking		7	الحجب النفسي- الصوتي
Public carriers		1	الناقلين العموميين
Public Key Infrastructure	PKI	8	بنية تحتية للتشفير بالمفاتيح العامة

Public Land Mobile Network	PLMN	6	شبكة البيت النقالة ذات الأرض العامّة
Public Switched Telephone Network	PSTN	6	شبكة الهاتف العمومية المحولة
Public-domain protocol		2	بروتوكول ذو ملكية عامة
Public-key certificates		8	شهادات تصديق المفاتيح العامّة
Public-key encryption		8	التشفير بالمفاتيح العامة
Pull protocol		2	بروتوكول سحب
Pulse Code Modulation	PCM	7	تضمين شفرة النبضات
Push protocol		2	بروتوكول دفع
Quality of Service	QoS	4,7	جودة الخدمة
Quantization		7	تكميم
Quantum Cryptography		8	التشفير الكمّي
Query flooding		2	فيضان الاستفسار، فيض الاستفسار
Queueing, Queuing		1,4	الانتظار في الصف (الطابور)، الاصطفاف
Queueing models		7	نماذج صفوف الانتظار
Queueing delay		2	زمن الانتظار في الصف (الطابور)
Quotas		9	حصص الاستخدام
Random Access Memory, Read And Write Memory	RAM	5	ذاكرة القراءة والكتابة
Random access protocols		5	بروتوكولات الوصول العشوائي
Random first selection		2	الاختيار العشوائي الأول
Random-early detection algorithm	RED	4	خوارزمية "الكشف المبكّر العشوائي"
Rarest first		2	الأندر أولاً
Real Time Protocol	RTP	7	بروتوكول الوقت الحقيقي
Real-time applications		2	تطبيقات الوقت الحقيقي (كإرسال الصوت والفيديو)

Real-time services		4	خدمات فورية، خدمات الوقت الحقيقي
Real-Time Streaming Protocol	RTSP	7	بروتوكول تشغيل المحتوى في الوقت الحقيقي
Reassembly		5	إعادة التجميع
Records		8	سجلات
Recursive query		2	الاستفسار التتابعي (كما في DNS)
Redirection		7	إعادة التوجيه
Redistributors		2	نقاط إعادة التوزيع
Redundancy reduction		7	تقليل البيانات الفائضة
References		7	إشارات مرجعية
Registrars		2	مسجِّلين
Relay agent		4	وكيل ترحيل
Relays		2	المُرحِّلات
Reliability		2	الاعتمادية
Reliable byte-stream channel		2	قناة لتدفق البايتات الموثوق
Reliable data transfer		2,3,5	النقل الموثوق للبيانات
Reliable data transfer protocol		3	بروتوكول نقل البيانات الموثوق
Reliable service		3	خدمة موثوقة
Remote directory		2	مجلد الملفات البعيد
Remote login		2	الوصول إلى الحاسبات عن بُعد (كاستخدام telnet)، الولوج عن بُعد
Rendezvous Point	RP	4	نقطة الالتقاء
Repeater		1,5	مكرّر
Replay attack		4	هجوم إعادة التشغيل
Replicated web servers		2	خادمات الويب المكررة

Request for Comments	RFC	1,2,3	طلب تعليقات
Request/response mode		9	نمط الطلب والرد
Request-To-Send	RTS	6	طلب إرسال
Reservation		7	الحجز
Residential access networks		1	شبكات الوصول السكني
Residential ISP		2	موفّر خدمة الإنترنت السكني (مثل AOL)
Resource Management	RM	3	إدارة الموارد
Resource over-provisioning		7	زيادة الرصيد الاحتياطي للموارد
Resource Records	RRs	2	سجلات الموارد
Resource reservation		7	حجز الموارد
Resource ReSerVation Protocol	RSVP	5,7	بروتوكول حجز الموارد
Resources		1	موارد
Reverse Path Forwarding	RPF	4	تمرير المسار العكسي
Rewind		7	الإعادة
RIP advertisements		4	إعلانات RIP
RIP updates		3	تحديثات RIP
Rivest- Shamir-Adleman algorithm	RSA	8	خوارزمية RSA
roBOT NETwork	Botnet	1,2	شبكة الروبوت (ويسيطر عليها الأشرار ويستخدمونها لاختراق الشبكات وإرسال رسائل الدعاية)
Root DNS servers		2	خادمات DNS الجذرية
Rotary telephones		4	الهواتف الدوّارة
Round Robin	RR	7	التعاقب الدوراني
Round Trip Time	RTT	3,7	زمن رحلة الذهاب والإياب
Route		1,4	مسار
Route oscillation		4	تذبذب المسارات

Routers		1,2	الموجّهات
Routing		4	توجيه
Routing algorithms		4	خوارزميات التوجيه
Routing Information Protocol	RIP	3,4	بروتوكول معلومات التوجيه
Routing loops		4	حلقات التوجيه، مسارات مغلقة
Routing processor		4	معالج التوجيه
Satellite		1	قمر صناعي، ساتل
Satellite networks		5	شبكات الأقمار الصناعية
Satellite radio spectrum		1	طيف ترددات الراديو للانتقال عبر القمر الصناعي
Scalability		6,7	قابلية التوسع
Scalar objects		9	كائنات قياسية
Scrambling function		8	عملية البعثرة، الخلط
Search engines		2	محركات البحث (مثل جوجل وياهوو)
Secure Hash Algorithm	SHA	8	خوارزمية التجزئة الآمنة
Secure Socket Layer	SSL	2,8	طبقة المقابس الآمنة (توفر الأمن في طبقة النقل لتوصيلات TCP)
Security		2	الأمن
Security Association	SA	8	ارتباط الأمن
Security Parameter Index	SPI	8	دليل متغير الأمن
Segmentation		5	تجزيء
Segments		3,5	قطع بيانات طبقة النقل، قطع (مقاطع) من الشبكات المحلية
Selective Repeat protocol	SR	3	بروتوكول "الإعادة الانتقائية"
Self-scalability		2	القدرة الذاتية على التوسع
Semaphore		3	سيمافور (إشارة تستخدم للتحكم في الوصول للموارد المشتركة)

Semi-permanent connections		4	توصيلات شبه دائمة
Sensors		1	مجسات، عناصر التحسس
Sequence		9	تسلسل، سلسلة
Sequence number		3,7	رقم تسلسلي، رقم مسلسل
Sequential search		4	بحث تتابعي (تعاقبي)، بحث خطي
Server farm (cluster of servers)		2	مزرعة خادمات (مجموعة من الخادمات)
Server		1,2	خادم، مخدم، ملقم، (سيرفر)
Service Level Agreements	SLAs	9	اتفاقيات ضمان مستوى الخدمة
Service pricing		7	تسعير الخدمة
Service providers		2	موفّرو الخدمة
Service Set IDentifier	SSID	6	معرّف مجموعة الخدمة
Session		3	جلسة
Session Initiation Protocol	SIP	7	بروتوكول إنشاء الجلسة
Set		9	مجموعة، طاقم
Shadow copy		4	نسخة ظلّ
Shadow fading		1	اضمحلال الإشارة وخفوتها بالحجب
Shared Ethernet		1	تقنية الإيثرنت المشتركة
Shared key		9	مفتاح مشترك
Shared-memory multiprocessor		4	معالجات متعددة ذات ذاكرة مشتركة
Shareware		8	برمجيات مجانية، برامج مجانية
Short Inter-Frame Spacing	SIFS	6	فترة المباعدة القصيرة بين الإطارات
Signaling		5	إرسال إشارات التحكم (التأشير)
Signaling messages		4,6	رسائل التحكم، رسائل التأشير

Signaling protocol		7	بروتوكول تأشير
Signal-to-Noise Ratio	SNR	6	نسبة الإشارة للضوضاء
Signature-based systems		8	أنظمة أساسها التوقيعات
Simple Gateway Monitoring Protocol	SGMP	9	البروتوكول البسيط لمراقبة البوابة
Simple Mail Transfer Protocol	SMTP	1,2	بروتوكول نقل رسائل البريد الإلكتروني البسيط
Simple Network Management Protocol	SNMP	3,9	بروتوكول إدارة الشبكة البسيط
Single-mode fiber optics		1	ألياف ضوئية (بصرية) وحيدة النمط
Slave node		6	عقدة ثانوية (التابع)
Sliding window protocol		3	بروتوكول النافذة المنزلقة
Slotted ALOHA protocol		5	بروتوكول ألوها الشرائحي
Slow Start	SS	3	البداية البطيئة
Small Office Home Office	SOHO	4	شبكات سوهو (المكتب الصغير والمكتب المنزلي)
Smoothed average		7	متوسط تنعيمي
Sniffers		8	برامج التقاط الرزم
Socket programming interface		2	واجهة برمجة المقابس
Sockets		2,3	مقابس
Soft guarantee		7	ضمان مرن
Soft state		4,7	حالة مرنة
Software		5	برمجيات
Software components		1	المكونات البرمجية
Source description packets		7	رزم وصف المصدر
Source identifier fields		7	حقول تعريف المصدر
Source quench		4	رسالة خنق المصدر

Source-Specific Multicast	SSM	4	الإرسال الجماعي المحدد بمصدر معين
Spam		1	رسائل بريد الدعاية الإلكتروني
Spanning tree		5	شجرة اتصال ممتدة
Spanning tree broadcast		4	الإذاعة عبر الشجرة الممتدة
Spatial redundancy		7	المعلومات الفائضة مكانياً
Speed of convergence		4	سرعة التقارب
Split-horizon connection		6	التوصيلة المنقسمة
Stacked headers		5	ترويسات MPLS المرصوصة
Star topology		1,5	طبوغرافية النجمة، طبولوجيا النجمة، الشكل النجمي للشبكة
State information		2	معلومات عن الحالة
Stateful Packet Filters		8	مرشحات الرزم ذات الحالة
Stateless protocol		2	بروتوكول بدون حالة
Statistical multiplexing		1	الإرسال المتعدد الإحصائي
Status		9	الحالة
Status code		2	رموز الحالة
Steady state		3	حالة الاستقرار
Stereo		7	صوت مجسم
Stop-and-wait protocol		3	بروتوكول التوقّف والانتظار
Storage disks		2	أقراص التخزين، اسطوانات التخزين
Store-and-forward		1,5	أسلوب التخزين والإرسال، أسلوب "خزّن ومرر"
Stream ciphers		8	شفرات التدفق
Structure of Management Information	SMI	9	هيكل معلومات الإدارة
Structured Query Language	SQL	3	لغة الاستفسارات المهيكلة (قواعد البيانات)

Stub network		4	شبكة عقب، شبكة طرف أو نهاية
Subfields		6	حقول فرعية
Subnet		4,5,6	شبكة فرعية
Super peers		2	نظائر ممتازة، نظائر "عليا"
Supercomputer		8	حاسب عملاق
Switch poisoning		5	تسميم المحوّل (نوع من الهجوم على الشبكة)
Switched Ethernet		1,5	تقنية الإيثرنت المحوّلة
Switches		1,2,5	محوّلات
Switching		4	تحويل
Switching fabrics		4	أنسجة التحويل
Symmetric session key		8	كمفتاح الجلسة المتماثل
Symmetric-key encryption		8	التشفير بمفاتيح متماثلة
Synchronization		7	تزامن
Synchronization Source Identifier	SSRC	7	معرّف مصدر التزامن
Syntax		9	الصيغة
Tabular structure		9	هيكل جدولي
Tag		7	علامة
TCP segment		2	قطعة TCP
TCP stack scans		8	مسح رصّة TCP
TCP/IP Protocol Suite		1	مجموعة بروتوكولات TCP/IP
Teardown		7	إنهاء
Teleconferencing		4	المؤتمرات عن بُعد
Temporal Key	TK	8	مفتاح مؤقت
Temporal redundancy		7	المعلومات الفائضة زمانياً
Terrestrial radio spectrum		1	طيف ترددات الراديو للانتقال الأرضي

Third Generation	3G	6	الجيل الثالث
Thread		3	عملية فرعية بسيطة، تشعيبة
Three-way handshake		4	مصافحة ثلاثية الاتجاه
Threshold		3,6	عتبة
Throughput		1,2,3,9	الطاقة الإنتاجية، معدل تدفق البيانات
Tier-1		1	الطبقة-1
Time frames		5	إطارات زمنية
Time slots		1	فترات أو شرائح زمنية
Time stamp		1,3,7	خاتم بقيمة الوقت الحالي، خاتم الوقت
Time-Division Multiplexing	TDM	1,5,6	الإرسال المتعدد بتقسيم الوقت، تجميع الإشارات بتقسيم الوقت، الإرسال المتعدد بتقسيم الزمن
Timeout		4	انتهاء الوقت
Timeout interval (for retransmission)		3	فترة الموقّت (لإعادة الإرسال)
Timers		3	الموقّتات
Time-to-Live	TTL	4,5,8	فترة العُمر
Timing		2	التوقيت
Token ring		5	حلقة العلامة
Token-passing		5	تمرير العلامة ("التوكن")
Token-ring protocol		5	بروتوكول حلقة العلامة
Top-Level Domain	TLD	2	نطاق المستوى الأعلى
Torrent		2	سيل
Traceroute		1,4	برنامج تتبع المسار، متتبع المسار
Tracker		2	المقتفي
Traffic conditioning		7	تكييف حركة المرور
Traffic load		4	حِمْل مرور البيانات

Traffic policing		7	تنظيم حركة مرور البيانات
Traffic shaping		7	تشكيل حركة المرور
Trailer		8	تذييل
Transaction		2	عملية
Transmission time		1	زمن الإرسال
Transmission Control Protocol	TCP	1,2,3,5	بروتوكول التحكم في الإرسال
Transport layer		1,3	طبقة النقل
Transport Layer Security	TLS	8	أمن طبقة النقل (نسخة معدلة بعض الشيء من الإصدار الثالث لطبقة المقابس الآمنة)
Transport mode		4	نمط النقل
Trap message		9	رسالة مصيدة
Triangle routing problem		6	مشكلة مثلث التوجيه
Triple-DES	3DES	8	معيار تشفير البيانات القياسي الثلاثي
Truncation attack		8	هجوم البتر
Tunnels		4	أنفاق
Twisted-pair copper wires	TP	1	أزواج أسلاك النحاس المجدولة
Type of Service	TOS	4	نوع الخدمة
Type, Length, Value	TLV	9	النوع، الطول، القيمة
Ubiquitous computing environments		6	بيئات حوسبة موجودة في كل مكان
Undirected graphs		4	الرسوم البيانية غير المتجهة
Unguided media		1	أوساط غير موجّهة
Unicast		4	إرسال فردي، إرسال أحادي (أي من نقطة إلى نقطة)
Unicast transmission		7	إرسال أحادي

Uniform Resource Locator	URL	1	محدد الموارد الموحد (طريقة معيارية لكتابة عناوين الموارد على الويب كعنوان صفحة ويب)
Universal Plug and Play Protocol	UPnP	4	بروتوكول "وصل وشغل" العام
Unreliable service		3,5	خدمة غير موثوقة
Unshielded Twisted Pair - Category 5	UTP 5	5	أسلاك نحاس مجدولة من الفئة الخامسة
Unshielded Twisted Pairs	UTPs	1	أزواج الأسلاك المجدولة المكشوفة
Unspecified Bit Rate	UBR	5	معدّل البتات غير المحدّد
Update		2	التحديث
Upgrade		2	ترقية ، توسعة
Uplink channel		5	قناة الوصلة الصاعدة
Uppercase letters		2	حروف كبيرة
Upstream		1	الاتجاه الصاعد إلى الإنترنت
User Datagram Protocol	UDP	1,3	بروتوكول وحدة بيانات المُستخدِم
Username		9	اسم المستخدم
Utilization		3	استغلال (مدى استغلال الموارد ، الاستغلالية)
Variable Bit Rate	VBR	5	معدل البتات المتغير
Very-high speed Digital Subscriber Line, Very-high bit rate Digital Subscriber Line	VDSL	1	خط المشترك الرقمي عالي السرعة
Video clip		2	مقطع فيديو
Video conferences, Video conferencing		2,7	مؤتمرات الفيديو
Video streaming		1	عرض شرائط الفيديو
View-based Access Control Model Configuration MIB		9	قاعدة بيانات لإدارة التهيئة بنموذج تحكم في الوصول مبني على المشهد

Virtual Circuit Identifier, Virtual Channel Identifier	VCI	3,4,5	معرّف الدائرة الافتراضية، معرّف القناة الافتراضية
Virtual Circuit, Virtual Channel	VC	3,9	دائرة افتراضية، قناة افتراضية
Virtual link		5	وصلة افتراضية
Virtual Private Network	VPN	4,5,8	شبكة افتراضية خاصة
Viruses		8	فيروسات
Visitor Location Register	VLR	6	سجل موقع الزوار
Voice mail box		7	صندوق البريد الصوتي
Voice over IP	VoIP	1,6	نقل الصوت على بروتوكول الإنترنت (كمكالمات الهاتف على الإنترنت)، مكالمة هاتفية عبر الإنترنت
Vulnerability points		8	نقاط الضعف
Warm start		9	البدء الساخن، الإقلاع الساخن
Web caches		1	ذاكرات الويب المخبأة
Web server		1	خادم ويب
Web, World-Wide Web	WWW	2	الويب (الوب)، شبكة الويب العالمية، شبكة المعلومات الدولية، الشبكة العنكبوتية
Webcam		7	كاميرا الويب
Weighted average		4	متوسط موزون
Weighted Fair Queuing	WFQ	7	سياسة الانتظار الموزون العادل
Wide-Area Network	WAN	1,4,5,6	شبكة واسعة النطاق، شبكة منطقة واسعة
Wideband Code Division Multiple Access	WCDMA	6	الوصول المتعدد عريض الحيز الترددي بتقسيم الكود
WiFi hotspots		6	بقع WiFi الساخنة (نقاط اتصال بالشبكة)
WiFi jungle		6	غابة WiFi

Wired Equivalent Privacy	WEP	6,8	الخصوصية المكافئة للشبكات السلكية
Wireless access networks		1	شبكات الوصول اللاسلكي
Wireless channel		6	قناة لاسلكية
WIreless FIdelity	WiFi	5,6	تقنية الدقة الرقمية (تستخدم المعايير IEEE 802.11)
Wireless links		5	وصلات لاسلكية
Wireless Local Area Networks	WLANs	1,5,6	شبكات محلية لاسلكية
Wireless Mesh Networks	WMNs	6	الشبكات اللاسلكية المشبكة، الشبكات اللاسلكية المعشقة
Wireless networks		6	الشبكات اللاسلكية
Wireless Personal Area Network	WPAN	6	شبكة اللاسلكي للمنطقة الشخصية
Wireless sensor network		6	شبكة المجسات اللاسلكية
Wireless station		6	محطة لاسلكية
Work conservation		7	حفظ العمل
Workload		2	حمل الشغل
Workstation		4	محطة عمل فرعية
World Interoperability for Microwave Access	WiMAX	1,6	تقنية التوافق العالمي للوصول عبر موجات المايكرويف (معيار IEEE 802.16)
World Wide Web	WWW	1	الويب، شبكة الويب العالمية، شبكة المعلومات الدولية، الشبكة العنكبوتية
Worms		8	ديدان

الملحق 3: وقفة مع ترجمة وتعريب المصطلحات

نظرا للجدل المثار حول قضية تعريب العلوم وغرابة المصطلحات واختلافها من بيئة لأخرى، آثارنا أن نوضح للقارئ بعض الأمور الهامة في هذا الملحق. فمن الجدير بالذكر أنه رغم هذا الجدل إلا أن هناك العديد من الجوانب الإيجابية للترجمة والتعريب والتي تتجاوز قضية المصطلحات والخلاف الدائر حول كيفية تعريبها (والذي يعيق في كثير من الأحيان مسيرة هذا العمل الجليل)؛ كسلاسة الأسلوب ووضوح العرض وبساطة التراكيب اللغوية لتوصيل المادة العلمية، والتوازن بين عدم التقيد الجامد باللفظ الأصلي ومراعاة الأمانة في النقل. وقد قمنا بتعريب المصطلحات في هذا الكتاب اعتماداً على التذوق اللغوي وعلى خبرتنا في المجال التقني للكتاب وبعد الرجوع لعدد من القواميس والمعاجم اللغوية الفنية المعتمدة كقاموس المورد، ومعجم المصطلحات العلمية (مجمع اللغة العربية بالقاهرة)، ومعجم الكيلاني لمصطلحات الكمبيوتر والإنترنت (للدكتور تيسير الكيلاني)، ودليل شعاع لمصطلحات الحاسب، ومعجم مصطلحات الكمبيوتر (الدار العربية للعلوم)، وبنك المصطلحات الموحدة (مكتب تنسيق التعريب بالرباط)، والبنك الآلي السعودي للمصطلحات العلمية (باسم)؛ وبالاطلاع على بعض ما نشر من الكتب العربية والموسوعات العلمية المتخصصة الأخرى؛ وبالاطلاع على عدد من مواقع الإنترنت كموقع الجمعية الدولية للمترجمين واللغويين العرب (واتا)، وموقع شبكة صوت العربية، وموقع الجمعية العلمية السورية للمعلوماتية، وموقع مجمع اللغة العربية الأردني (تعريب المصطلحات العلمية)، وموقع المركز العربي للتعريب والترجمة والنشر بدمشق (مجلة التعريب)، وموقع الموسوعة العربية، وموقع مجلة عالم الكمبيوتر والإنترنت، وغيرها من مواقع الإنترنت الأخرى؛ وأحياناً بالتشاور المتأني وبالاستعانة بمدقق لغوي. ومن الضوابط التي راعيناها – قدر المستطاع – في انتقاء المصطلح العربي: مكافأته للمصطلح الأصلي وعدم التباسه مع مصطلح آخر وشيوعه وقصره وخفة لفظه ومطابقته لقواعد اللغة.

بشكلٍ عام هناك خمسة مذاهب رئيسة متعارف عليها في ترجمة المصطلحات. فبعض المترجمين يكتب المصطلح كما يُنطق (وهو ما يعرف بـ transliteration). فمثلاً يستخدمون كلمة "ماوس" لترجمة "mouse"، ويستخدمون "بت" لترجمة "bit"، ويستخدمون "إنترنت" لترجمة "Internet"، ويستخدمون "فيديو" لتعني "video"، وهكذا. وآخرون يستخدمون المرادف باللغة العربية للمصطلح فمثلاً يستخدمون "فأرة" لترجمة "mouse"

ويستخدمون "رقم ثنائي" لترجمة "bit"، وهكذا. والمذهب الثالث يحاول صياغة ألفاظ ومصطلحات باللغة العربية كمقابل للمصطلح الأجنبي (وهو ما يعرف بالتحوير)؛ فمثلاً تُستخدم "الأشعة السينية" لترجمة "x-ray"، وتُستخدم "الشابكة" لترجمة "Internet"، وهكذا. وهناك من يفضل أن يكتب المصطلح كما هو بلغته الأصلية بدون تعريب. وهناك من يكتب المعنى الوصفي ("التراكيب الوصفية للمعنى الوظيفي") للمصطلح كأن تستخدم "مشيرة" لترجمة "mouse" وتقول "شبكة عالمية للربط بين الشبكات التي تستخدم مجموعة البروتوكولات TCP/IP للاتصال فيما بينها....." لترجمة "Internet". يلاحظ أن هذه المصطلحات تتفاوت فيما بينها لاختيار الطريقة الأنسب لترجمتها. وسبب ذلك الخلاف أن المصطلح يرتبط في الذهن بتصور معين والذي قد يعتمد على بيئة القارئ وسياق الحديث وبيئة المنشأ للمصطلح ومدى انتشاره. وهذا أمر خلافي حتى في المصطلحات غير التقنية؛ مثلاً ماذا يعني "لمس النساء"؟ وماذا تعني "حَبْحَبْ"؟ وغير ذلك الكثير.

فمثلاً ماذا نترجم الكلمات "Java"، "Google"، "Aloha"، "Cookies"، الخ. إن مثل هذه الكلمات تعتبر كأسماء الأعلام ارتبطت بتصور معين ويجب أن تترك كما هي. خاصةً وقد أصبحت تلك المصطلحات في كثير من الأحيان دارجة ومعروفة حتى لرجل الشارع، وأنه يتكرر ذكرها عشرات بل مئات المرات في الكتاب!! فهل يستساغ أن نترجم "Java" على أنها "نوع من البن"، ونترجم "Cookies" على أنها "كعك"، ونترجم "Internet" في كل مرة ترد في الكتاب على أنها "شبكة عالمية تربط بين الشبكات التي تستخدم مجموعة البروتوكولات TCP/IP للاتصال فيما بينها وتتألف من مجموعة من الأجهزة وخطوط النقل......"؟! أو أن نقصد مباشرة لما ارتبط بالذهن بقول "Java" أو "جافا"، وقول "كوكيز" أو "Cookies"، وقول "Internet" أو "إنترنت"!

الخلاصة: إنّنا لا ندعي أننا سنحل هذا الخلاف، لكننا راعينا المزج بين المذاهب المختلفة حفاظاً على سلاسة الأسلوب وسلامة التراكيب اللغوية ومراعاةً لمدى شيوع اللفظ في البيئة العربية حتى نتجنب استخدام ألفاظ مستغربة (قدر المستطاع) والتي قد يعيق أو يصعب من فهم المادة العلمية. وقد تم ذلك بالرجوع لعدد من مجامع اللغة العربية ومعاجم المصطلحات كما بينا سابقاً وبالاطلاع على بعض الكتب العربية في مجال الكتاب، وأحياناً بالاجتهاد والتشاور المتأني في بعض الألفاظ المستحدثة. فالهدف من الكتاب التقني

هو توصيل المعلومة بشكل سريع وسلس للقارئ مع مراعاة قواعد اللغة ما أمكن ذلك (دون افتعال لغوي). وجدير بالذكر أن عدد المصطلحات التي يوجد خلاف في ترجمتها ليس كبيراً.

ولنأخذ عينة لترجمة بعض المصطلحات كما هو مبين في الجدول التالي:

المصدر	Bit	Byte	Internet	Web	Real time	Computer
مجمع اللغة العربية بالقاهرة	بيتة	بايت، ثمانية أرقام ثنائية، جزء من كلمة الحاسب	الإنترنت، الشبكة الدولية	الويب، الشبكة العنكبوتية العالمية	الوقت الحقيقي، الوقت الفعلي	الحاسب
بنك المصطلحات الموحدة، مكتب تنسيق التعريب، المنظمة العربية للتربية والثقافة والعلوم، المغرب	ثنائية	ثُمانية	- -	- -	الزمن الحقيقي، الآني	حاسوب، حاسب
البنك الآلي السعودي للمصطلحات (باسم)، السعودية	بت، بتة، رقم ثنائي، شارة، خانة	بايت، شارة	الإنترنت، الشبكة العنكبوتية العالمية	الشبكة العنكبوتية العالمية.	الوقت الحقيقي، الوقت الفعلي، الوقت الآني، فوري	الحاسب الآلي، آلة حاسبة، كمبيوتر

معجم الكيلاني لمصطلحات الكمبيوتر والإنترنت (للدكتور تيسير الكيلاني وهو فلسطيني قام بالتدريس في عدد من البلدان العربية)، مكتبة بيروت 2004	بت، رقم ثنائي، نبضة، شارة	بايت، مجموعة أرقام ثنائية، ثماني بتات	الإنترنت	الويب، نصوص الشبكة العالمية للاتصالات	الزمن الحقيقي	كمبيوتر، جهاز حاسب، الحاسب الآلي، الحاسب الإلكتروني
معجم مصطلحات الكمبيوتر، الدار العربية للعلوم، 2001، لبنان	بت، خانة ثنائية	بايت، ثماني بتات	الإنترنت	Web	الزمن الحقيقي	حاسوب، كمبيوتر
دليل شعاع لمصطلحات الحاسب، طبعة 2004، سوريا	بت، خلية ثنائية	بايت	الإنترنت	الويب	الزمن الحقيقي	حاسب
مترجم جوجل	مقدار ضئيل	بايت	الإنترنت	الإنترنت	الوقت الحقيقي	كمبيوتر
مترجم الوافي الذهبي	قطعة	بايت	الإنترنت	الويب	الوقت الحقيقي	حاسوب

الترجمة التي رجحناها لتلك المصطلحات

Bit	Byte	Internet	Web	Real time	Computer
بت	بايت	الإنترنت	الويب	الزمن الحقيقي	الحاسب

ونود أن نشير إلى أنه رغم تبني أحد المصطلحات إلا أننا في بعض الأحيان أشرنا لبعض البدائل الأخرى (وخصوصاً عندما يذكر المصطلح لأول مرة وعندما نجد لذلك ضرورة لتوصيل المعنى بشكل أفضل).

نذكر مثالاً آخر لمصطلحات قريبة في المعنى إلا أنها تعني أشياء مختلفة عندما تستخدم المصطلح الأصلي؛ مثال ذلك المصطلحات: label، token، flag، semaphore، signal. وقد استخدمنا ألفاظاً مختلفة للتفريق بينها كما يلي:

signal	semaphore	flag	token	Label
إشارة	سيمافور	علم	علامة، توكن	وسمة

في النهاية ينبغي التأكيد على أننا لسنا في هذا الكتاب بصدد إعداد معجم موحد للمصطلحات وإنما آثارنا التبسيط ووضوح وسلاسة العرض وتسلسل الأفكار لتوصيل المعنى العلمي المراد عند الوقوف على قضية المصطلح.